

L Number	Hits	Search Text	DB	Time stamp
1	2147	707/3.ccls.	USPAT	2003/09/05 16:21
2	0	707/104.ccls.	USPAT	2003/09/05 16:21
3	1349	707/2.ccls.	USPAT	2003/09/05 16:22
4	1056	707/101.ccls.	USPAT	2003/09/05 16:22
-	0	sgrep	USPAT	2003/08/22 10:47
-	103	grep	USPAT	2003/08/22 10:47
-	693	otsubo	USPAT	2003/08/22 10:48
-	0	otsubo AND motohide	USPAT; JPO	2003/08/22 10:48
-	0	otsubo AND motohide	JPO	2003/08/22 10:48
-	4253	otsubo	JPO	2003/08/22 10:49
-	17	otsubo AND editing	JPO	2003/08/22 10:51
-	314	715/530.ccls.	USPAT	2003/08/22 15:48
-	2	grep AND redirect	USPAT	2003/08/22 16:02
-	421	edit AND "search results"	USPAT	2003/08/22 16:03
-	37	edit SAME "search results"	USPAT	2003/08/22 16:22
-	5	"wild card" SAME delimited	USPAT	2003/08/22 16:29
-	658	negation AND search	USPAT	2003/08/22 16:30
-	38	negation SAME search	USPAT	2003/08/25 11:24
-	1	("6259421").PN.	USPAT	2003/08/25 11:24
-	0	("6259421").PN.	JPO	2003/08/25 11:24
-	0	6-259421	JPO	2003/08/25 11:25
-	0	"6259421"	JPO	2003/08/25 11:25
-	0	"6-259421"	JPO	2003/08/25 11:25
-	0	"6-259421"	DERWENT	2003/08/25 11:43
-	0	"6-259421"	JPO	2003/08/25 11:43
-	0	"6259421"	JPO	2003/08/25 11:43
-	2	("5812999").PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/05 16:10
-	2	("5649218").PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/05 16:11

Search History 09/357,841

STIC Translation Branch Request Form forPhone: 308-0881 Crystal Plaza ¼, Room 2C15 <http://ptoweb/patents/stic>**PTO 2003-5172**

S.T.I.C. Translations Branch

Information in shaded areas marked with an * is required**Fill out a separate Request Form for each document***U. S. Serial No. : 09/353,841*Requester's Name: Jonathan SchreiberPhone No.: 305-9777Office Location: 4806, CPK2Art Unit/Org. : 2178Is this for the Board of Patent Appeals? NoDate of Request: 8/25/2003*Date Needed By: 9/29/2003

(Please indicate a specific date)

Document Identification (Select One):Note: If submitting a request for patent translation, it is not necessary to attach a copy of the document with the request.If requesting a non-patent translation, please attach a complete, legible copy of the document to be translated to this form and submit it at your EIC or a STIC Library.1. ☒ **Patent***Document No. 6-259421*Country Code JP*Publication Date 9-16-94*Language Japanese

No. of Pages _____ (filled by STIC)

2. _____ **Article**

*Author _____

*Language _____

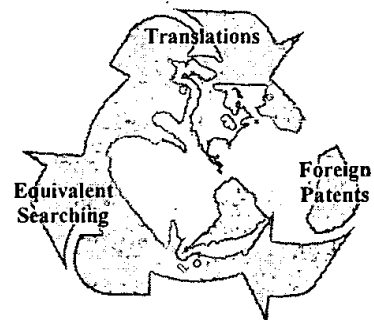
*Country _____

3. _____ **Other**

*Type of Document _____

*Country _____

*Language _____

Translations Branch
The world of foreign prior art to you.

To assist us in providing the most cost effective service, please answer these questions:

Will you accept an English Language Equivalent? _____ (Yes/No)

Would you like to review this document with a translator prior to having a complete written translation?

(Translator will call you to set up a mutually convenient time) _____ Yes/No

Would you like a Human Assisted Machine translation? _____ (Yes/No)

Human Assisted Machine translations provided by Derwent/Schreiber is the default for Japanese Patents 1993 onwards with an Average 5-day turnaround.

Copy of HMAT E-mail 9.2.03**STIC USE ONLY****Copy/Search**Processor: SJSDate assigned: 8/25Date filled: 8/25Equivalent found: (Yes/No) (Yes)

Doc. No.: _____

Country: _____

TranslationDate logged in: 8.16.03

PTO estimated words: _____

Number of pages: 117

In-House Translation Available: _____

In-House

Translator: _____

Assigned: _____

Returned: _____

Contractor:Name: ALW

Priority: _____

Sent: 8-16-03Returned: 9-1-03

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-259421

(43)公開日 平成6年(1994)9月16日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/20	5 5 0 E	7315-5L		
	5 5 4 M	7315-5L		
15/40	5 0 0 L	9194-5L		

審査請求 未請求 請求項の数7 O L (全 33 頁)

(21)出願番号 特願平5-164761

(22)出願日 平成5年(1993)7月2日

(31)優先権主張番号 特願平4-176792

(32)優先日 平4(1992)7月3日

(33)優先権主張国 日本(JP)

(31)優先権主張番号 特願平5-2855

(32)優先日 平5(1993)1月11日

(33)優先権主張国 日本(JP)

(71)出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂三丁目3番5号

(72)発明者 古城 慎太郎

神奈川県川崎市高津区坂戸3丁目2番1号

K S P R & D ビジネスパークビル

富士ゼロックス株式会社内

(72)発明者 宮崎 淳

神奈川県川崎市高津区坂戸3丁目2番1号

K S P R & D ビジネスパークビル

富士ゼロックス株式会社内

(74)代理人 弁理士 木村 高久

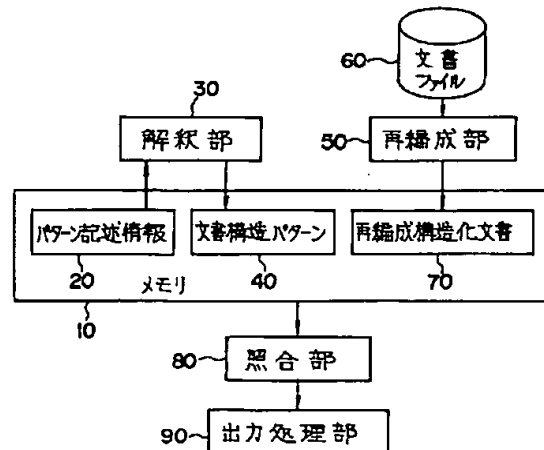
最終頁に続く

(54)【発明の名称】 文書処理装置

(57)【要約】

【目的】構造化文書に対する指定された階層構造パターンに従った文書構成要素の検索処理を行うことのできる文書処理装置を提供する。

【構成】メモリ10には、文書構成要素同志の接続関係が記述された情報を示すパターン記述情報20が格納され、また文書ファイル60には構造化文書が格納されている。解釈部30は、パターン記述情報20を解釈して、階層構造を表現している文書構造パターンを生成し、この文書構造パターン40をメモリ10に格納する。再編成部50は、文書ファイル60の構造化文書を走査して、照合処理可能な形式の構造に再編成し、この結果である再編成構造化文書70をメモリ10に格納する。照合部80は文書構造パターン40と再編成構造化文書70とを照合する。出力処理部90は、照合部80の照合により一致したものを出力する。



【特許請求の範囲】

【請求項1】複数の文書構成要素を有する構造化文書に対する処理を行う文書処理装置において、基準となる文書構成要素同志の接続関係を解釈する解釈手段と、

前記構造化文書を照合処理可能な形式の構造に再編成する再編成手段と、

前記解釈手段による解釈結果と、前記再編成手段による再編成結果とを照合する照合手段と、

この照合手段の照合により一致した文書構成要素を前記再編成結果から抽出し出力する出力処理手段とを具備したことを特徴とする文書処理装置。

【請求項2】所定の文書構成要素に対する所定の処理を指定する指定手段を更に具備し、

前記出力処理手段は、前記照合手段の照合により一致した文書構成要素に対して、前記指定手段により指定された所定の処理を施した後、出力することを特徴とする請求項1記載の文書処理装置。

【請求項3】所定の文書構成要素に対する削除処理を指定する指定手段と、

前記照合手段の照合により一致した文書構成要素と、当該文書構成要素の親である文書構成要素とを対応付けして蓄積する蓄積手段とを更に具備し、

前記出力処理手段は、前記指定手段により削除処理が指定されると、前記蓄積手段に蓄積されている前記親である文書構成要素から、前記蓄積手段に蓄積されている前記照合手段の照合により一致した文書構成要素に関する情報を取り除くと共に、当該親である文書構成要素から削除されない、当該親の文書構成要素の子供である文書構成要素を出力することを特徴とする請求項1記載の文書処理装置。

【請求項4】複数の文書構成要素を有する構造化文書を複数保存したファイルを格納する格納手段と、

基準となる文書構成要素同志の接続関係を解釈する解釈手段と、

前記格納手段に格納されているファイル内の複数の構造化文書それぞれを照合処理可能な形式の構造に再編成する再編成手段と、

前記解釈手段による解釈結果と、前記再編成手段による再編成結果とを照合する照合手段と、

この照合手段の照合により一致した文書構成要素を前記再編成結果から抽出し出力する出力処理手段とを具備したことを特徴とする文書処理装置。

【請求項5】前記複数の構造化文書を保存した複数のファイルを対象として、前記文書構成要素同志の接続関係に適合する文書構成要素を前記再編成結果から抽出することを特徴とする請求項4記載の文書処理装置。

【請求項6】ソース側及びターゲット側それぞれの前記複数の構造化文書を保存した複数のファイルを対象として、前記文書構成要素同志の接続関係に適合する文書構

成要素を前記再編成結果から抽出すると共に、前記ターゲット側の抽出結果である文書構成要素に対する前記ソース側の抽出結果である文書構成要素の挿入を実行することを特徴とする請求項4記載の文書処理装置。

【請求項7】前記出力処理手段は、指定された属性に関する情報に基づいて、前記照合手段の照合により一致した文書構成要素の属性の参照又は変更の操作を実行することを特徴とする請求項2乃至請求項6記載の文書処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、文書中から所望のパターンを検索する文書処理装置に関する。

【0002】

【従来の技術】従来においては、ワードプロセッサや、文書作成・編集機能を有するワークステーションやコンピュータ等の装置には、一般的に、作成された文書中から所望の文字列を検索することのできる検索機能が設けられている。この検索機能を利用することにより、検索対象となる文字列を、他の文字列に置換したり削除したりして、文書を編集することができる。

【0003】またこのような文書編集を自動化するようにしたものも実現されており、例えば、

(1) カット・アンド・ペーストしながらの操作をマクロ記述して何度でも実行することができるようにしたものの。

(2) 例えば特開平3-147062号公報に開示されている様に、文書中から取り出された複数の文字列を一時記憶領域に保存し、その後順次、ペースト（挿入）するようにしたもの。

(3) 指定したパターン文字列が文字列中に現れた際に、置き換え等を実施する様にしたもの（例えばUNIXのsedのようなストリームエディタ）。がある。

【0004】また文書ファイルを編集する方法としては、インタラクティブにユーザがカット・アンド・ペーストする方法や、バッチ処理で行う方法がある。このうち、効率良く複数回の抽出挿入を行う方法として、例えば、

(4) ユーザが指定した抽出文字列を複数個同時に記憶しておき、これら複数の文字列を順次、所定の位置に挿入するようにしたもの（例えば特開平3-147062号公報）。

(5) ユーザがインタラクティブに文書にマークを付与することにより、一度に複数の指定領域の入れ替えを行うようにしたもの（例えば特開平3-260761号公報）。

(6) ファイル内のある特定の文字列パターンの出現に対して、文字列の置き換え等の処理を行うようにしたもの（例えばUNIXのawk、sedなど）。

(7) 構造と内容とを分離して管理する構造化文書シス

テムにおいて、差し込み指定のある複数の文書から、差し込み文字列を予め抽出しておき、この差し込み文字列を差し込み対象文書へ反映させるようにしたもの（例えば特開平4-57151号公報）。がある。

【0005】更に文書の特定部分の属性値（例えば文字の大きさなど）を変更するようにしたものも実現されており、この方法として、例えば、

（8）インタラクティブなエディタにより逐一変更するようにしたもの。

（9）特定部分にスタイルを予め設定しておき、スタイルの属性値を変更することにより、一括変換するようにしたもの。

がある。

【0006】

【発明が解決しようとする課題】しかしながら、上記従来の装置では、文書中から文字パターン（検索対象の文字列）を検索することは可能であるが、これは文書中の文字列と指定文字列との照合により一致したものを検索するようにしているので、内部的に階層構造の情報を持った文書いわゆる構造化文書から、指定された階層構造パターンに従った文書要素を検索することができなかった。

【0007】すなわち、構造化文書においては、文書要素が階層構造上のどこに位置するかによって、その文書要素の情報は意味を変えるので、従来の如く、単に文字列の照合のみにより検索し、階層構造を考慮しない検索では、適切な部分にアクセスすることができず、正確な検索処理を行うことができなかった。

【0008】また上記（1）のものでは、不正確な処理を施してしまうことが多く、このため、複数の文書をバッチで処理することができない。

【0009】また上記（2）のものでは、正確な処理を施すことはできるが、バッチで処理することができない。

【0010】また上記（3）のものでは、バッチ処理に適しているが、構造を持った文書の文字列は構造上の位置によって意味を変えるので、不適切な部分を書き換えてしまう恐れがあり、構造化文書には適していない。

【0011】また上記（4）のものでは、文書中の全指定箇所をユーザがインタラクティブに一度設定しなければならないので、操作が複雑である。

【0012】また上記（5）のものでは、上記（4）の場合と同様に、文書中の全指定箇所をユーザがインタラクティブに一度設定しなければならないので、操作が複雑である。

【0013】また上記（6）のものでは、ユーザが文書中の所定箇所を指定する必要はないが、一般的なストリングマッチのため、ファイル中の構造の意味を解釈せずに、単に文字列として照合し、一致した文字列に対して変更を加えるようにしているので、文書の構造の意味を

維持しつつ処理することはできない。

【0014】また上記（7）のものでは、内容の変更にについては効率良く実行することができるが、構造と内容を同時に変更することはできない。

【0015】また上記（8）の方法では、インタラクティブに処理するので、属性値の変更処理を自動化することができず、このため効率が悪く、操作ミスによる間違いも発生し易い。

【0016】更に上記（9）のものでは、予めスタイルを設定しておく必要であるので、そのための手間がかかる。また同一のスタイルには全て同一の属性が付与されるので、柔軟性に欠ける。

【0017】そこで本発明の目的は、構造化文書に対する指定された階層構造パターンに従った文書構成要素の検索処理を行うことができる文書処理装置を提供することである。

【0018】本発明の他の目的は、構造化文書に対する指定された階層構造パターンに従った文書構成要素を検索し、この文書構成要素に対する削除、置換、複写などの操作を実施することができる文書処理装置を提供することである。

【0019】本発明の他の目的は、複数の構造化文書に対する指定された階層構造パターンに従った文書構成要素の検索処理を行うことができる文書処理装置を提供することである。

【0020】本発明の他の目的は、ターゲット側の複数のファイルそれぞれに保存されている複数の構造化文書から抽出された文書構成要素に対する、ソース側の複数のファイルそれぞれに保存されている複数の構造化文書から抽出された複数の文書構成要素の挿入操作を一度に自動的に行うことができる文書処理装置を提供することである。

【0021】本発明の他の目的は、複数の構造化文書に対する指定された階層構造パターンに従った文書構成要素を検索し、この文書構成要素の属性値の参照又は変更を行うことができる文書処理装置を提供することである。

【0022】

【課題を解決するための手段】第1の発明は、複数の文書構成要素を有する構造化文書に対する処理を行う文書処理装置において、基準となる文書構成要素同志の接続関係を解釈する解釈手段と、前記構造化文書を照合処理可能な形式の構造に再編成する再編成手段と、前記解釈手段による解釈結果と、前記再編成手段による再編成結果とを照合する照合手段と、この照合手段の照合により一致した文書構成要素を前記再編成結果から抽出し出力する出力処理手段とを具備している。

【0023】第2の発明は、第1の発明において、所定の文書構成要素に対する所定の処理を指定する指定手段を更に具備し、前記出力処理手段は、前記照合手段の照

合により一致した文書構成要素に対して、前記指定手段により指定された所定の処理を施した後、出力することとを特徴としている。

【0024】第3の発明は、第1の発明において、所定の文書構成要素に対する削除処理を指定する指定手段と、前記照合手段の照合により一致した文書構成要素と、当該文書構成要素の親である文書構成要素とを対応付けして蓄積する蓄積手段とを更に具備し、前記出力処理手段は、前記指定手段により削除処理が指定されると、前記蓄積手段に蓄積されている前記親である文書構成要素から、前記蓄積手段に蓄積されている前記照合手段の照合により一致した文書構成要素に関する情報を取り除くと共に、当該親である文書構成要素から削除されない、当該親の文書構成要素の子供である文書構成要素を出力することとを特徴としている。

【0025】第4の発明は、複数の文書構成要素を有する構造化文書を複数保存したファイルを格納する格納手段と、基準となる文書構成要素同志の接続関係を解釈する解釈手段と、前記格納手段に格納されているファイル内の複数の構造化文書それぞれを照合処理可能な形式の構造化文書に再編成する再編成手段と、前記解釈手段による解釈結果と、前記再編成手段による再編成結果とを照合する照合手段と、この照合手段の照合により一致した文書構成要素を前記再編成結果から抽出し出力する出力処理手段とを具備している。

【0026】第5の発明は、第4の発明において、前記複数の構造化文書を保存した複数のファイルを対象として、前記文書構成要素同志の接続関係に適合する文書構成要素を前記再編成結果から抽出することとを特徴としている。

【0027】第6の発明は、第4の発明において、ソース側及びターゲット側それぞれの前記複数の構造化文書を保存した複数のファイルを対象として、前記文書構成要素同志の接続関係に適合する文書構成要素を前記再編成結果から抽出すると共に、前記ターゲット側の抽出結果である文書構成要素に対する前記ソース側の抽出結果である文書構成要素の挿入を実行することとを特徴としている。

【0028】第7の発明は、第2の発明乃至第6の発明において、前記出力処理手段は、指定された属性に関する情報に基づいて、前記照合手段の照合により一致した文書構成要素の属性の参照又は変更の操作を実行することとを特徴としている。

【0029】

【作用】第1の発明では、解釈手段によって解釈された文書構造パターンと、再編成手段によって再編成された構造化文書とを照合手段によって照合し、更に出力処理手段が、その照合により一致した文書構成要素を再編成された構造化文書から抽出し出力するようにしたので、構造化文書から、指定された階層構造に従った文書構成

要素を検索し出力することができる。

【0030】第2の発明では、第1の発明において、出力処理手段は、照合手段の照合により一致した文書構成要素に対して、指定手段により指定された所定の処理例えば削除、置換、複写などの処理を施した後、出力するようにしているので、構造化文書から、指定された階層構造に従った文書構成要素に対して、削除、置換、複写などの処理を施すことができる。

【0031】第3の発明では、第1の発明において、出力処理手段は、指定手段により削除処理が指定されると、蓄積手段に蓄積されている、照合手段の照合により一致した文書構成要素の親である文書構成要素から、蓄積手段に蓄積されている照合手段の照合により一致した文書構成要素に関する情報（例えば文書構成要素を示すノード、そのノードの位置情報）を取り除くと共に、当該親である文書構成要素から削除されない、当該親の文書構成要素の子供である文書構成要素を出力するようにしているので、削除すべき文書構成要素の親の文書構成要素の内容を自動的に変更することができる。

【0032】第4の発明では、解釈手段が、基準となる文書構成要素同志の接続関係を解釈し、また再編成手段が、格納手段に格納されているファイル内の複数の構造化文書それぞれを照合処理可能な形式の構造化文書に再編成し、また照合手段が、解釈手段による解釈結果と、再編成手段による再編成結果とを照合し、更に出力処理手段が、照合手段の照合により一致した文書構成要素を前記再編成結果から抽出するようにしているので、複数の構造化文書から、指定された階層構造に従った文書構成要素を検索し出力することができる。

【0033】第5の発明では、第4の発明において、複数の構造化文書を保存した複数のファイルを対象として、文書構成要素同志の接続関係に適合する文書構成要素を再編成結果から抽出するようにしているので、複数のファイルそれぞれに保存されている複数の構造化文書から、指定された階層構造に従った文書構成要素を検索し出力することができる。

【0034】第6の発明は、第4の発明において、ソース側及びターゲット側それぞれの複数の構造化文書を保存した複数のファイルを対象として、文書構成要素同志の接続関係に適合する文書構成要素を再編成結果から抽出すると共に、ターゲット側の抽出結果である単数又は複数の文書構成要素に対するソース側の抽出結果である単数又は複数の文書構成要素の挿入を実行するようにしているので、ターゲット側における複数のファイルそれぞれに保存されている複数の構造化文書から抽出された単数又は文書構成要素に対して、ソース側における複数のファイルそれぞれに保存されている複数の構造化文書から抽出された単数又は文書構成要素を一度に挿入することができる。

【0035】第7の発明では、第2の発明乃至第6の発

明において、出力処理手段は、指定された属性に関する情報に基づいて、照合手段の照合により一致した文書構成要素の属性の参照又は変更の操作を実行するようにしている。構造化文書から、指定された階層構造に従った文書構成要素を検索し、この文書構成要素の属性に対する参照又は変更の操作を実施することができる。

【0036】

【実施例】以下、第1の実施例乃至第5の実施例を添付図面を参照して説明する。

【0037】最初に第1の実施例を図1乃至図9を参照して説明する。

【0038】図1は、本発明に係る文書処理装置の第1の実施例を示す機能ブロック図である。

【0039】同図において、メモリ10には、基準となる文書構成要素同志の接続関係（階層関係や順序関係）のパターン記述情報20（これについては後述する）が記憶されており、解釈部30は、メモリ10からパターン記述情報20を読み出して解釈し、この解釈結果である文書構造パターン40（これについては後述する）をメモリ10に記憶する。再編成部50は、文書ファイル60に保存されている構造化文書内を走査して、その構造化文書を照合処理可能な形式の構造に再編成し、この再編成結果である再編成構造化文書70（これについては後述する）をメモリ10に格納する。そして照合部80は、メモリ10に記憶されている文書構造パターン40と再編成構造化文書70と照合し、この照合結果を出力処理部90に出力する。出力処理部90では、照合部*

（節／表題／introduction）＃（節／表題） …（1）

ここで、“／”は包含関係、“＃”は順序関係を表している。

のように記述される。この記述内容はメモリ10に記憶される。

【0044】そして解釈部30は、メモリ10から上記（1）に示す様なパターン記述情報を読み出して解釈し、この結果として図3に示す様に階層構造（木構造）を形成している文書構造パターン（これが上述した文書構造パターン40に相当する）を生成する。このとき、当然、上述したような構文要素や文法などが考慮されて、文書構造パターンが生成されることとなる。なおこの実施例では、図3に示すような文書構造パターンにおける矩形図形を単純パターンということにする。

【0045】ここで、解釈部30によるパターン記述の解釈処理について、図4に示すフローチャートを参照して説明する。なおここでは、括弧（“（”、“）”）の構文要素の処理を省略している。

【0046】解釈部30は、カレントレコードを生成し（ステップ401）、その後、入力文字列（例えば上記（1）のパターン記述情報）についての解釈は終りか否かを判断する（ステップ402）。入力文字列についての解釈処理がまだ残っている場合は、次の文字が順序関※50

*80の照合により一致した文書構成要素を再編成構造化文書70から抽出し出力する。

【0040】図2は、図1に示した実施例の装置を実現するためのハードウェア構成を示したものであり、例えば、ワークステーションやコンピュータ等のブロック図を示している。図2において、装置は、構造化文書に対する検索処理を実行する中央処理装置（以下、CPUという）210と、主メモリ220と、ディスク230と、各種のデータ内容や文書内容を表示するディスプレイ240と、キーボードやマウスから構成され各種データや指令を入力する入力装置250とがバス260を介してそれぞれ接続されている。なおCPU210はバス260を介してこれに接続された各部を制御する。

【0041】ここで、図1に示した機能ブロック図の構成要素と図2に示したブロック図の構成要素との対応関係について説明する。図1に示したメモリ10は主メモリ220に対応しており、図1に示した解釈部30、再編成部50、照合部80、および出力処理部90は共にCPU210に対応しており、文書ファイル60はディスク230に対応している。

【0042】次に上述したパターン記述情報20について説明する。

【0043】パターン記述情報20は、パターンが、“節”、“表題”などの単純文字列パターン、“／”、“＃”などの接続表現記号、“（”、“）”などの括弧、等の構文要素が特定の文法に従って出現するように表現されるものであり、例えば、

※係を表す記号“＃”か否かを判断する（ステップ403）。

【0047】ステップ403において記号“＃”であれば、新たなレコードを生成し、このレコードをカレントレコードの弟にし（ステップ404）、その後、新たに生成したレコードをカレントレコードと定義する（ステップ405）。その後、上記ステップ402に戻りこのステップ以降を実行する。

【0048】ステップ403において文字が記号“＃”でない場合は、当該文字が包含関係を表す記号“／”か否かを判断し（ステップ406）、記号“／”の場合は、新たなレコードを生成し、このレコードをカレントレコードの子にする（ステップ407）。その後、ステップ405に進む。

【0049】ステップ406において文字が記号“／”でない場合は、文字であることを意味するので、当該文字をカレントレコード内に挿入し（ステップ408）、その後、ステップ402に戻りこのステップ以降を実行する。

【0050】なお、ステップ402において入力文字列についての解釈が終了した場合は処理を終了する。

【0051】ここで、具体例を挙げて説明する。例えば

“富士夫／太郎＃花子”という文字列は、図5(a)～(j)に示すようにパターンとして解釈されていく。なお、同図において、矩形図形がレコードを表している。また図5(j)に示す内容が最終的な文書構造パターンである。

【0052】上述した例では、文字列のパターン解釈であったが、図形エディタを用いて描画したグラフ(グラフ理論におけるグラフ)を解釈してパターンとする方法もある。このときは、ノードやリンクを適切な意味に解釈するように定義する。例えば、図3に示した例では、矩形で囲まれた文字列(例えば節や表題)が単純文字列パターンを表し、上下の矩形図形を結んでいるリンク(例えば符号301で示す線分(リンク))が序列関係を表し、左右の矩形図形を結んでいるリンク(例えば符号302で示す線分(リンク))が包含関係を表している。このような図形から意味構造を抽出するには、例えばパターン記述専用の図形エディタを用意すれば良い。

【0053】ここで、図形エディタを用いたパターン記述の一例を図6に示す。

【0054】まずユーザは、図6(a)に示す様にパターンエディタの初期画面つまりウィンドウ600を表示画面に表示させ、次に図6(b)に示す様にノード“unspecified node”をマウス(入力装置250に設けられている)を操作して選択し、その後、所定の操作を行って、図6(c)に示す様にポップアップメニュー610を表示させる。そして、ポップアップメニュー610の“set string”の項目を選択して、図6(d)に示す様に文字列を記述する。続いて、ポップアップメニュー610の“make child”の項目を選択して、図6(e)に示す様にノード“節”の子ノードを作成する。引き続き、ポップアップメニュー610の“make brother”の項目を選択して、図6(f)に示す様に弟ノード作成する。こうして作成された図6(f)に示す様なグラフは直接文書構造パターンとして用いられる。すなわち、図6(f)に示す内容が、パターン記述情報20であり、また文書構造パターン40でもある。

【0055】次に、再編成部50により再編成される再編成構造化文書70について説明する。

【0056】ファイルとして保管されている文書のままでは、文書内部の構造へのアクセスができずパターン処理に不利なので、ファイルを走査して構造を再編成する。ただし、一度にファイル全体を解析する必要はなく、照合部80が必要するとき、必要となっている部分のみを解析して出力するようにする。

【0057】この解析処理としては、ファイルの必要とする箇所にファイルポインタを移動して可変長レコードを切り出し、そのレコードに予め記述されている構造上での位置情報をもとに木構造(或いは部分木構造)を再構成するようになっている。再編成して得られた再編成構造化文書の一例を図7に示す。この図に示す様に文書

は、階層構造(木構造)として表現される。なおこの実施例では、再編成構造化文書における矩形図形を文書ノードということにする。

【0058】次に、照合部80による照合処理について、図8に示すフローチャートを参照して説明する。

【0059】照合部80は、current-nodeを、再編成構造化文書の構造における最初の文書ノードにし(ステップ801)、current-pat を、文書構造パターンの構造における最初の単純パターンにする(ステップ802)。

【0060】その文書ノードおよび単純パターンは指定された接続条件に一致するか否かを判断し(ステップ803)、一致する場合は、current-pat とcurrent-nodeとが一致するか否かを判断する(ステップ804)。

【0061】ステップ804において一致する場合は、current-nodeを次の文書ノードにし(ステップ805)、その後、単純パターンが終りか否かを判断する(ステップ806)。

【0062】ステップ806において単純パターンが終了した場合は、その旨が照合部80から出力処理部90に通知される。出力処理部90では、その一致した文書ノードを出力する(ステップ807)。このようにして出力される文書ノードは、ディスプレイ240に表示されたり、あるいはファイルとして保管される。さらには、その文書ノードを他の(或いは同一の)文書中の特定領域に流し込むことによって、文書内容を編集することができる。このように検索して一致した文書ノードすなわち文書構成要素を、削除や置換したり、他の文書に挿入することができる。

【0063】ところでステップ807が終了すると、照合部80は、current-pat を前の単純パターンにし(ステップ808)、その後、文書ノードが終りか否かを判断す(ステップ809)。

【0064】文書ノードが終了したら処理を終了し、まだ文書ノードがある場合は上記ステップ803に戻りこのステップ以降を実行する。

【0065】なお、ステップ806において単純パターンが終りの場合はステップ809に進む。

【0066】また上記ステップ803において指定された接続条件に不一致の場合、ステップ804において一致しない場合は、current-pat は最初の単純パターンか否かを判断し(ステップ810)、単純パターンであれば、一致していた文書ノードまで戻り(ステップ811)、その後、ステップ805に進む。

【0067】ステップ810において単純パターンでない場合はcurrent-pat を前の単純パターンにし(ステップ812)、その後、ステップ811に進む。

【0068】以上のような処理を行って得られた照合結果を図9に示す。この図に示した例では、図3に示した文書構造パターンと図7に示した再編成構造化文書とを

照合した場合の結果を示している。

【0069】この例においては、図9中点線で示されるように、単純パターン901Aと文書ノード901Bとが一致し、単純パターン902Aと文書ノード902Bとが一致し、単純パターン903Aと文書ノード903Bとが一致し、単純パターン904Aと文書ノード904Bとが一致し、単純パターン905Aと文書ノード905Bとが一致し、単純パターン906Aと文書ノード906Bとが一致している。

【0070】この図から分かるように、文書構造パターン（階層構造情報）が分かれば、例えば、単純パターン906Aの文字列が分からない場合であっても、照合処理することにより、その単純パターン906Aに対応する文書ノード906Bを得ることができ、その結果として、表題は「魔神の宅配便」であるということが分かる。

【0071】また単純パターン903Aの文字列「introduction」に一致するところが、文書ノード903B、907に存在していた場合であっても、必ず表題になっているもののみしか一致しないので（この例では文書ノード903Bのみ一致）、確実に検索することができる。

【0072】以上説明したように第1の実施例によれば、文字列パターンに加えて、適切な方法で文書の構成要素の接続関係を示すことによって、誤りなく必要とするものが得られる。

【0073】次に第2の実施例を図10乃至図15を参照して説明する。

【0074】図10は、本発明に係る文書処理装置の第2の実施例を示す機能ブロック図である。この機能ブロック図は、図1に示した第1の実施例の機能ブロック図の構成において、適合ノード蓄積部1010、命令処理部1020を追加し、出力処理部90を出力処理部1030に変更した構成になっている。なお図10にお*

（節／本文段落／スタミナX）#注

ここで、／は包含関係を示す記号

#は順序関係を示す記号

^は処理対象となるノードを示す記号

が記述されメモリ10に記憶されている。

【0083】次に解釈部30は、図4に示した第1の実施例のパターン解釈処理手順と同様の処理を実行して、上記（2）に示すパターン記述情報20から、図11に示す様な文書構造パターンを生成し、これを文書構造パターン40としてメモリ10に格納する。このとき処理対象となるノードは「注」とであると解釈する。図11では、処理対象となるノードの目印として二重枠で囲んで表記している。この場合も、図5に示した第1の実施例の具体例の様にパターン解釈されていく。

【0084】この第2の実施例でも、図形エディタを用いて描画したグラフ（グラフ理論におけるグラフ）を解※50

*いて、図1に示した構成要素と同様の機能を果たす部分には同一の符号を付している。

【0075】適合ノード蓄積部1010は、照合部80の照合結果である文書構成要素を蓄積する。

【0076】命令処理部1020は、コマンドラインあるいは標準入力から与えられる所定の処理を解釈し、この解釈結果を出力処理部1030に与える。なお所定の処理には、“挿入する”、“置換する”、“削除する”の処理が含まれている。

【0077】出力処理部1030は、適合ノード蓄積部1010に蓄積されている文書構成要素に対して、命令処理部1020からの処理命令に従って処理を実行し、出力する。この出力は、次の処理のための標準出力に出力しても良い。

【0078】なお上記標準入力及び標準出力とは、UNIX（オペレーティングシステム）における標準入力及び標準出力のことである。

【0079】この図10に示した装置も、図2に示した第1の実施例のハードウェア構成で実現することができる。ここで図10に示した構成要素と図2に示した構成要素との対応関係について説明する。図10に示した適合ノード蓄積部1010は図2に示した主メモリ220に対応し、図10に示した命令処理部1020及び出力処理部1030は共に図2に示したCPU210に対応している。他の構成要素については第1の実施例と同様である。

【0080】この第2の実施例は、基本的には第1の実施例と同様である。第1の実施例と異なるのは、構造化文書中から、文書構造パターンに一致する構造を抽出し、この抽出した構造に対して、“挿入”、“置換”、“削除”などの処理を施すという点である。

【0081】そこで、第2の実施例における文書編集処理について、図11乃至図15を参照して説明する。

【0082】パターン記述情報20として、

…（2）

※釈して文書構造パターンを認識することができる。図形エディタを用いたパターン記述の方法は、図6を用いて説明した第1の実施例と同様である。

【0085】一方、再編成部50による再編成処理結果は、図12に示す内容であり、メモリ10に再編成構造化文書70として格納される。

【0086】そして照合部80が、図8に示した第1の実施例の照合処理手順と同様の処理を実行して、図11に示す文書構造パターンと、図12に示す再編成構造化文書とを照合する。この結果として、図13に示すような照合結果が得られることとなる。図13においては、単純パターン1310と文書ノード1310A、1310Bとが一致し、単純パターン1320と文書ノード1320A、1320Bとが一致し、単純パターン1330と文書ノード1330A、1330Bとが一致し、単

純パターン1340と文書ノード1340A、1340Bとが一致している。このとき照合部80は、パターン記述情報中で指示されている処理対象ノードに対応する構造化文書中のノードを適合ノード蓄積部1010に格納する。この例では「注」のノード1340A、1340Bが蓄積される。なおノードの全情報を格納する必要はなく、ノードへのリンクを記憶するようにしても良い。

【0087】命令処理部1020に、例えば「insert “スタミナXは絶倫製薬の登録商標です” as lastChild」というコマンドラインが与えられると、命令処理部1020は、そのコマンドラインを、出力処理部1030の動作を決定する変数として解釈し、この解釈結果に応じた処理を実行する。この処理を図14を用いて説明する。

【0088】最初にinsertを解釈して、“挿入する”を示すinsertのコマンドを出力処理部1030に設定する。コマンドの設定は、解釈したコマンドを設定するようにしても良いし、出力処理部1030に予め設定されたコマンド群を用意しておき、解釈したコマンドに該当するものを設定するようにしても良い。ここでは、後者の方法を採用しており、“挿入する”、“置換する”、“削除する”にそれぞれ対応するコマンドinsert、replace、deleteが予め用意されている。これらのコマンド以外にも、insert from file“tottekoi”，insert from stdin、removeなどのコマンドを用意することもできる。

【0089】次に“スタミナXは絶倫製薬の登録商標です”を解釈して、処理上必要となるデータ領域（ここではbufferという名前参照される）に文字列を出力処理部1030に複写する。

【0090】最後にas lastChildを解釈して、末子であるということを示すフラグ1を出力処理部1030に設定する。

【0091】出力処理部1030では、「insertのコマンド」、「buffer=スタミナXは絶倫製薬の登録商標です」、「フラグ1」が設定されると、これらの情報に基づいて、適合ノード蓄積部1010に蓄積されているノードに対して処理を施し、この結果をファイルストリームに出力する。この出力結果を図15に示す。この図15に示す例では、注のノード1340A、1340Bの子供として“スタミナXは絶倫製薬の登録商標です”のノード1500A、1500Bが接続されている。なお処理と出力は同時に実行することができる。更には対象ノードの照合の検証とも同時に実行することができる。

【0092】以上説明したように第2の実施例によれば、構造化文書内のオブジェクト間の関係を利用したパターン、つまり基準となる階層構造のパターンとのパターンマッチングを行うようにしているので、構造化文書における正確な情報（文書構成要素）にアクセスすることができると共に、階層上のデータ（文書構成要素）の

位置を簡単に指定することができる。

【0093】また1つのパターンを使用して上述したようなパターンマッチングを行うようにしているので、構造化文書中の複数の書き換え又は挿入位置を指定することができる。

【0094】更にユーザとのインタラクションなしで正確なアクセスを実行することができるので、構造を持った文書のバッチ処理が可能となる。

【0095】次に第3の実施例について、図16乃至図20を参照して説明する。

【0096】図16は、本発明に係る文書処理装置の第3の実施例を示す機能ブロック図である。この機能ブロック図は、図10に示した第2の実施例の機能ブロック図の構成において、適合ノード蓄積部1010を削除し、削除情報蓄積部1040を追加した構成になっている。

【0097】削除情報蓄積部1040は、照合部80の照合により一致した文書ノードと、当該文書ノードの親である文書ノードとを対応付けして蓄積（この蓄積の構造については後述する）し管理する。

【0098】出力処理部1030は、命令処理部1020から“削除する処理”が渡されると、削除情報蓄積部1040に蓄積されている照合部80の照合により一致した文書ノードの親である文書ノードから、削除情報蓄積部1040に蓄積されている照合部80の照合により一致した文書ノードに関する情報（例えば文書ノード、そのノードの位置情報）を取り除くと共に、当該親の文書ノードから削除されない子供の文書ノードを出力する。

【0099】なお構造化文書においては、一般的に、各ノード間の関係を表現する際に、親ノードには自己の子供である子ノードを示す情報が含まれており、一方、子ノードに自己の親である親ノードを示す情報が含まれているので、出力処理部1030は、“削除する処理”を受け取ったときは、親の文書ノードから、削除すべき文書ノードを示す情報を削除するようにしている。このとき、削除される文書ノードに含まれている親の文書ノードを示す情報は削除してもしなくとも良い。但し、親の文書ノードから削除すべき文書ノードを示す情報を削除することにより、当該削除すべき文書ノードは、親の文書ノードとの関連性がなくなり出力されないため、処理効率の点からいって、削除される文書ノードに含まれている親の文書ノードを示す情報は削除しない方が良い。

【0100】図17は、削除情報蓄積部1040に蓄積される削除情報の構造の一例を示している。この実施例では、削除情報の構造を、削除される文書ノード（以下、子ノードという）のリスト（以下、子リストという）を持った、その親の文書ノード（以下、親ノードという）のリスト（以下、親リストという）として表現するようにしている。因みに図17においては、親リスト

には、2つの子ノードC1-1、C1-2の子リストを持つ親ノードP1と、2つの子ノードC2-1、C2-2の子リストを持つ親ノードP2とが登録されている。

【0101】なお、上記の例では削除される子ノードを登録するようにしているが、削除される子ノードの情報としては、何番目の子供が削除されるべきか、という情報で十分である。

【0102】次に、削除情報蓄積部1040による削除情報の作成処理について、図18に示すフローチャートを参照して説明する。

【0103】削除情報蓄積部1040は、初期化として親リストを空にし（ステップ1051）、次に、照合部80から渡される照合結果つまりパターンと一致するノード（以下、これをノードCとする）を順次受け取ると共に、当該ノードCを1つ取り込む（ステップ1052）。

【0104】次に削除情報蓄積部1040は、取り込んだノードCの親ノードPは親リストに未登録か否かを判断する（ステップ1053）。

【0105】ステップ1053において親ノードPは登録済みの場合は、ノードCは親ノードPの子リストに未登録か否かを判断する（ステップ1054）。

【0106】ステップ1054においてノードCは登録済みの場合は、照合部80からの次のノードCを取り込む（ステップ1055）。

【0107】なお、ステップ1054においてノードCが未登録の場合は、ノードCを親ノードPの子リストに新規登録し（ステップ1056）、その後、ステップ1055に進み、またステップ1053において親ノードPが未登録の場合は、親リストに新規登録し（ステップ1057）、その後、ステップ1054に進む。

【0108】ところで上記ステップ1055を終了した場合、削除情報蓄積部1040は、照合部80から渡されるノードは終りか否かを判断し（ステップ1058）、ノードが終りの場合には処理を終了し、一方、まだノードが存在している場合は、上記ステップ1053に戻り、照合部80から渡されるノードが終了するまで、ステップ1053～1058を繰り返す。

【0109】ここで、図17に示した例を用いて、削除情報の作成処理を説明する。

【0110】図17において、親ノードP1が親ノードP2より早く出現するものとし、今現在、親リストは空き状態とする。

【0111】このような状態で、上記ステップ1053において、子ノードC1-1が取り込まれた後、ステップ1053が実行された際には、子ノードC1-1の親ノードP1は未登録であるので、この場合はステップ1057に進み、このステップにより親ノードP1が親リストに登録される。このステップ1057終了後はステップ1054に移行するが、子ノードC1-1は親ノードP1

の子リストには登録されていないので、ステップ1056に進み、このステップにより子ノードC1-1が親ノードP1の子リストに登録される。

【0112】そしてステップ1055、1058が実行されることとなり、この場合は、照合部80からのノード（つまり照合結果であるノード）がまだ存在するので、上記ステップ1053に戻り、このステップにより子ノードC1-2についての処理が実行される。この場合は、ステップ1053においては「NO」（つまり親ノードP1は親リストに登録済み）となるので、ステップ1054に進み、このステップにおいては「YES」（子ノードC1-2は親ノードP1の親リストには未登録）なので、ステップ1056に進み、このステップにより子ノードC1-2が親ノードP1の親リストに登録される。

【0113】以下同様に、子ノードC2-1、C2-2についての処理が行われる。

【0114】なお図17において、削除すべきノードが、子ノードC1-1、C1-2、親ノードP1であった場合は、親リストに、例えば親ノードP1が削除される旨の情報のみを登録し、子ノードC1-1、C1-2については登録しないようにする。何故ならば、削除されるノードから更に削除されるということは無いので、親ノードP1が削除される旨のみを登録すれば良いこととなる。従って、子ノードC1-1、C1-2のリストへの登録を行う必要がないので、処理効率及び記憶使用効率を向上させることができる。

【0115】次に、出力処理部103の出力処理について、図19に示すフローチャートを参照して説明する。

【0116】出力処理部103は、命令処理部1020からの“削除する処理”を受け取ると（ステップ1061）、処理の対象をファイル先頭に移動し（ステップ1062）、その後、そのファイルから、文書のルートノード（これはファイル先頭）であるノードNを1つ読み取り（ステップ1063）、該ノードNについて、出力すべきノードを出力するノード出力処理を実行する（ステップ1064）。すなわちステップ1064においては、ノードNの下位に存在するノードが出力されることになる。

【0117】このステップ1064のノード出力処理について、図20に示すサブルーチンを参照して説明する。

【0118】出力処理部103は、ノードNは親リストに登録されているか否かを判断する（ステップ1071）。ここで、ノードNはルートノードであるので、このノードNが、親リストに登録されているということは、必ず削除されるノード（すなわち子ノード）が存在することを意味しており、一方、親リストに登録されていない場合は、削除されるノードが存在しないことを意味している。

【0119】ところで、ステップ1071において登録済みの場合は、ノードNについて変更を施して出力する(ステップ1072)。このステップ1072においては、削除情報蓄積部1040に蓄積されている親リストつまり削除情報に基づいて、ノードNについて、親リストに登録されている親ノードから、該親ノードの子リストに登録されている子ノードに関する情報(例えば子ノードを示す情報)を取り除く。この処理が終了した後は、ノードNには出力すべきノードのみが存在していることになる。

【0120】そしてステップ1072を終了した後、出力処理部103は、ノードNについての子リストに含まれない子をノード(つまり出力すべきノード)に対して、再帰的にノード出力処理を実行する(ステップ1073)。

【0121】一方、ステップ1071において未登録の場合は、ノードNをそのまま出力し(ステップ1074)、そのノードNの子ノード全てに対して、再帰的にノード出力処理を実行する(ステップ1075)。

【0122】以上説明したように第3の実施例によれば、パターン照合して削除されるノード(ノードC)が検出されると、このノードCと該ノードCの親のノード(ノードP)とを対応して記憶し、そして、親のノードPからノードCに関する情報(ノードCを示す情報)を取り除いて、その親のノードPを出力するようにしているので、ユーザが、親のノードPの内容を変更する必要がない。このため構造化文書に対する編集処理の作業効率を向上させることができる。

【0123】次に第4の実施例を図21乃至図27を参照して説明する。

【0124】図21は本発明に係る文書処理装置の第4の実施例を示す機能ブロック図である。同図において、文書処理装置は、メモリ1610、解釈部1620、再編成部1630、ソース文書ファイル群1640、ターゲット文書ファイル群1650、照合部1660、複数抽出点蓄積部1670、複数挿入点蓄積部1680、出力処理部1690、抽出挿入関係ルール指定部1700を備えている。

【0125】メモリ1610には、第1の実施例で説明したようなパターン記述情報1611が記憶されており、解釈部1620は、メモリ1610からパターン記述情報1611を読み出して解釈し、この解釈結果を第1の実施例で説明したような文書構造パターン1612としてメモリ1610に記憶する。

【0126】再編成部1630は、ソース文書ファイル群1640に保持されている複数の構造化文書内を走査して、これらの構造化文書を照合処理可能な形式の構造に再編成し、この結果をソース再編成構造化文書群1613としてメモリ1610に格納する。同様に、ターゲット文書ファイル群1650内の複数の構造化文書

についても、ソース再編成構造化文書群1614としてメモリ1610に格納する。

【0127】この第4の実施例において、構造化文書とは、章、節といった文書構造と文書内容とを一緒に持つフォーマットによる文書表現を意味している。1つのファイル内に複数の構造化文書を含むとは、図22に示す様に、1つのファイル内に、異なる論理根を持つ論理構造が複数含まれることを示している。論理根が異なる論理構造は互いに独立であり、且つ部分構造が共有されることはない。部分構造とは、構造化文書における一部分の構造のことであり、例えば図22中点線で囲まれた構造化文書においては、「節」というノード以下の構造や、「表題」というノード以下の構造などである。

【0128】ソース文書ファイルとは、部分構造を他の文書へ挿入するために、文書構造パターンに適合する部分構造が抽出される文書ファイル(つまりソース側の文書ファイル)のことである。結果としてこの文書ファイルの内容は変更されることはない。

【0129】一方、ターゲット文書ファイルとは、他の文書からの部分構造を挿入するために、文書構造パターンに適合する部分構造が抽出される文書ファイル(つまりターゲット側の文書ファイル)のことである。結果としてこの文書ファイルの内容は変更される。

【0130】またソース再編成構造化文書とは、ソース文書ファイル内の構造化文書に対する再編成処理の結果である再編成構造化文書のことであり、

【0131】一方、ソース再編成構造化文書とは、ターゲット文書ファイル内の構造化文書に対する再編成処理の結果である再編成構造化文書のことであり、

【0132】照合部1660は、複数ファイル抽出点認識部1661と、複数ファイル挿入点認識部1662とを有している。複数ファイル抽出点認識部1661は、ソース構造化文書群をそれぞれ格納する複数のファイルに対して、文書構造パターンによる照合によりファイルを走査し、複数の部分構造の抽出点を認識し、この認識結果を複数抽出点蓄積部1670に格納する。このときファイル名と抽出点の対の情報を格納する。一方、複数ファイル挿入点認識部1662は、ターゲット構造化文書群をそれぞれ格納する複数ファイルに対し、文書構造パターンによりファイルを走査し、複数の部分構造に対する挿入点を認識すると共に、この認識結果を複数挿入点蓄積部1680に格納する。このときファイル名と挿入点の対の情報を格納する。なお複数抽出点蓄積部1670と複数挿入点蓄積部1670とは独立しているが、抽出点の情報と挿入点の情報を区別するようにして、これらの情報を1つの蓄積部に蓄積するようにしても良い。

【0133】抽出挿入関係ルール指定部1700は、抽出点と挿入点との対応関係を、ファイルを跨がる1対1、あるいはファイルを跨がる複数n対1のいずれかの

10

20

30

40

50

ルールを出力処理部1690に与える。

【0134】出力処理部1690には、複数ファイル部分構造抽出挿入部1691が設けられており、複数ファイル部分構造抽出挿入部1661は、抽出挿入関係ルール指定部1700から与えられた抽出点と挿入点との対応関係の情報に基づいて、複数抽出点蓄積部1670に蓄積されている抽出点に対応する部分構造から、複数挿入点蓄積部1680に蓄積されている挿入点に対応する部分構造分への文書構造の挿入操作を実行する。

【0135】この図21に示した装置も、図2に示した第1の実施例のハードウェア構成で実現することができる。ここで、図21に示した機能ブロック図の構成要素と図2に示したブロック図の構成要素との対応関係について説明する。図26に示したメモリ1610は図2に示した主メモリ20に対応し、図21に示した解釈部1*

(節/表題/まとめ) #本文段落

ここで、/は包含関係を示す記号

#は順序関係を示す記号

が記述されメモリ1610に記憶されている。

【0139】次に解釈部1630によって、図4に示す第1の実施例のパターン解釈処理手順が実行されることにより上記(3)のパターン記述情報が解釈され、更にこの結果が文書構造パターン1612としてメモリ1610に記憶される。

【0140】続いて再編成部1930によって、ソース文書ファイル群1640とターゲット文書ファイル群1650とが再編成され、更にこれらの結果が、ソース再編成構造化文書群1613、ターゲット再編成構造化文書群1614としてメモリ1610に記憶される。

【0141】続いて照合部1660の複数ファイル抽出点認識部1661による抽出点認識処理について、図23を参照して説明する。図23はその処理動作を示すフローチャートである。

【0142】複数ファイル抽出点認識部1661は、最初のソース文書ファイル(ソース再編成構造化文書群1613中の1つのファイル)をメモリ1610から読み込んで(ステップ1801)、ソース文書ファイルは終りか否かを判断し(ステップ1802)、終りの場合には処理を終了し、一方、終りでない場合は、ファイル内の全ての構造化文書(つまり論理根を持つ文書)に対する処理が終了したか否かを判断する(ステップ1803)。

【0143】ここで、まだ未処理の構造化文書が存在している場合は、その構造化文書に対するパターン照合処理を実行し(ステップ1804)、その照合処理結果である抽出点を複数抽出点蓄積部1670に蓄積する(ステップ1805)。

【0144】上記ステップ1803において、全ての構造化文書について処理した場合は、次のソース文書ファイルをメモリ1610から読み込み、その後、上記ステ

*620、再編成部1630、照合部1660、出力処理部1690及び抽出挿入関係ルール指定部1700は共に図2に示したCPU210に対応し、ソース文書ファイル群1640及びターゲット文書ファイル群1650は共に図2に示したディスク230に対応している。

【0136】この第4の実施例も、基本的には第1の実施例と同様である。第1の実施例と異なるのは、1つのファイル内の複数の構造化文書に対して、文書構造パターンに一致する構造を抽出する点である。また複数の構造化文書を有するファイルを複数設け、これらのファイル内の複数の構造化文書に対して照合する点も異なっている。

【0137】そこで、第4の実施例における文書編集処理について、図23乃至図27を参照して説明する。

【0138】パターン記述情報20として、

…(3)

※アップ1802に戻る。

【0145】なおステップ1804のパターン照合処理は、図8に示す第1の実施例の処理手順と同様である。

【0146】同様に、複数ファイル挿入点認識部1662は、ターゲット文書ファイル(ターゲット再編成構造化文書群1614)に対する挿入点の認識処理を実行する。この結果は、複数挿入点蓄積部1680に蓄積される。

【0147】すなわち、複数ファイル抽出点認識部1661と複数ファイル挿入点認識部1662は基本的には同様の処理を実行し、異なるのは、対象となる文書ファイル(構造化文書)がソースであるかターゲットであるかという点である。

【0148】ここで、抽出点の認識処理結果の様子を図24に示す。図24において、ファイル1、ファイル2は、ソース再編成構造化文書を示しており、またハッチングの掛った部分が、文書構造パターン1612に適合した部分である。この図24から分かるように、ファイル内の複数の構造化文書及び複数のファイルに跨って、構造がパターンマッチングされ適合されている。この例での抽出点は、ハッチングの掛った部分の「節」というノードの直前の位置(つまり「論理根」というノードとの接続点の位置)である。この抽出点は、各ファイル毎に抽出点の列として複数抽出点蓄積部1670に蓄積される。

【0149】同様に挿入点の認識処理結果も、図24に示す様に、文書構造パターン1612に適合した部分が認識されることとなる。挿入点についても上記同様に考えることができる。

【0150】以上の説明から分かるように、この第4の実施例においては、図24に示すように、文書構造パターン1612に適合する部分構造(ハッチング部分)を抽出することが、本来の目的ではなく、「節」というノードを抽出することが目的なのである。しかし、図24

21

に示されるように、「節」というノード以下の構造には各種の部分構造が接続されているので、所望の「節」というノードを抽出するために、文書構造パターン1612との照合を実施しているのである。

【0151】次に、出力処理部1690の複数ファイル部分構造抽出挿入部1691の出力処理について、図25を参照して説明する。図25はその処理動作を示すフローチャートである。

【0152】複数ファイル部分構造抽出挿入部1691は、複数抽出点1蓄積部1670から各ファイル毎の抽出点の列を得る。これらをA[i] = (file名、抽出点)に順に格納すると共に(ステップ2001)、複数挿入点蓄積部1680から各ファイル毎の挿入点の列を得る。これらをB[j] = (file名、挿入点)に順に格納する(ステップ2002)。

【0153】次に、挿入抽出関係ルール指定部1700から指定された抽出点と挿入点との対応関係のルールが“ファイルを跨がる1対1”であるか否かを判断する(ステップ2003)。

【0154】ここで、“ファイルを跨がる1対1”の場合は、i = 1、j = 1と定義し(ステップ2004)、A[i]あるいはB[j]が終りか否かを判断する(ステップ2005)。

【0155】ここで、終りでない場合は、A[i]に示される抽出点に基づいて、ソース文書ファイル群から部分構造を抽出すると共に(ステップ2006)、この部分構造をB[j]に示される挿入点に挿入する(ステップ2007)。

【0156】その後、i = i + 1、j = j + 1と再定義した後(ステップ2008)、上記ステップ2005に戻る。ステップ2005においてA[i]あるいはB[j]が終りの場合は、結果を出力する(ステップ2009)。

【0157】上記ステップ2003においてルールが“ファイルを跨がる1対1”でない場合は、ルールが“ファイルを跨がる複数n対1”であるか否かを判断する(ステップ2010)。そうであれば、j = 1と定義し(ステップ2011)、その後、B[j]が終りであるか否かを判断する(ステップ2012)。

【0158】ここで、終りの場合は、A[1] ~ A[n]に示される抽出点に基づいて、ソース文書ファイル群から部分構造を全て抽出し、これらA[1] ~ A[n]までの部分構造を兄弟として繋ぐと共に(ステップ2013)、兄弟として繋がれた構造を、B[j]に示される挿入点に挿入する(ステップ2014)。この挿入点に対して、兄、弟、子供として挿入することができる。

【0159】上記ステップ2014を終了した後はj = j + 1と再定義し(ステップ2015)、その後、上記ステップ2012に戻る。すなわち結果として、B

22

[1] ~ B[n]の各挿入点に、兄弟として繋がれたA[1] ~ A[n]までの部分構造が挿入される。

【0160】上記ステップ2012においてB[j]が終了した場合は上記ステップ2009に進む。

【0161】上記ステップ2010においてルールが“ファイルを跨がる複数n対1”でない場合は挿入処理は行わない(ステップ2016)。

【0162】なお複数ファイル部分構造抽出挿入部1691は、ソース文書ファイル群の抽出点、ターゲット文書ファイル群の挿入点のいずれかの数が多いときは挿入処理を行わず、ステータスを返す。

【0163】例えば、抽出点の数 > 挿入点の数、のときステータスの値が1
抽出点の数 < 挿入点の数、のときステータスの値が2
この結果として、出力処理部1690からは、エラー通知が出力される。

【0164】また、抽出点の数 = 挿入点の数、のときは0のステータスを返す。この結果として、出力処理部1690からは、変更後のターゲット文書ファイルが出力されることとなる。

【0165】ここで、ファイルを跨がる1対1のルールに基づく挿入結果の様子を図21に示し、またファイルを跨がる複数n対1のルールに基づく挿入結果の様子を図22に示す。

【0166】以上説明したように第4の実施例によれば、複数の文書ファイルであって、且つ1つのファイル中に複数の構造化文書文書が保存されている場合であっても、構造化文書内のオブジェクト間の関係を利用したパターン、つまり基準となる階層構造のパターンとのパターンマッチングを行うようにしているので、構造化文書における正確な情報(文書構成要素)にアクセスすることができると共に、階層上のデータ(文書構成要素)の位置を簡単に指定することができる。

【0167】すなわち、ターゲット側の複数のファイルそれぞれに保存されている複数の構造化文書から抽出された文書構成要素に対する、ソース側の複数のファイルそれぞれに保存されている複数の構造化文書から抽出された文書構成要素の挿入操作を一度に自動的に行うことができるということである。

【0168】次に第5の実施例を図28乃至図31を参照して説明する。

【0169】図28は本発明に係る文書処理装置の第5の実施例を示す機能ブロック図である。この機能ブロック図は、図1に示した第1の実施例の機能ブロック図の構成において、ファイル位置情報保持部2310、属性情報指定部2320を追加し、出力処理部90を出力処理部2330に変更した構成になっている。なお図23において、図1に示した構成要素と同様の機能を果たす部分には同一の符号を付している。

【0170】ファイル位置情報保持部2310は、照合

部80の照合結果である文書構成要素のファイル内の位置情報を保持する。

【0171】属性情報指定部2320は、文書構成要素の属性情報を指定するものであり、属性値の参照のときは属性名を指定し、属性値の変更のときは属性名及び属性値を指定する。

【0172】出力処理部2330は、ファイル位置情報保持部2310に保持されている文書構成要素のファイル内の位置情報と、属性情報指定部2320から指定される属性情報とに基づいて出力処理を実施する。ここで、属性値の参照のときは、該当する文書構成要素の属性名を持つ属性の属性値を出力し、一方、属性値の変更のときは、該当する文書構成要素の属性名を持つ属性を、指定された属性値に変更して出力する。

【0173】なおここでは構造化文書は、図29(a)に示す様に各文書構成要素をノードとする木構造を持っているが、ファイル上では、図29(b)に示す様に、決められた規則に従って各文書構成要素は1列に並んでいる。また文書構成要素内の属性名と属性値の対も1列に並んでいるとする。

【0174】図28に示した装置も、図2に示した第1の実施例のハードウェア構成で実現することができる。ここで図28に示した機能ブロック図の構成要素と図2に示したブロック図の構成要素との対応関係について説明する。図28に示したファイル位置情報保持部2310は図2に示した主メモリ20に対応し、図28に示した属性指定部2320及び出力処理部2330は共に図2に示したCPU210に対応している。他の構成要素については第1の実施例と同様である。

【0175】この第5の実施例も、基本的には第1の実施例と同様である。第1の実施例と異なるのは、構造化文書中から、文書構造パターンに一致する構造を抽出し、この抽出した構造に対して、属性の参照又は変更の処理を施すという点である。

【0176】なおこの第5の実施例においては、照合部80による文書構造パターンと再編成構造化文書との照合処理までは、第1の実施例で説明した処理と同様なので、ここではその説明を省略し、属性の参照又は変更処理について説明する。

【0177】次に、文書処理装置の属性の参照又は変更処理について、図30及び図31を参照して説明する。図30は属性の参照処理動作のフローチャートを示し、図31は属性の変更処理動作のフローチャートを示している。

【0178】最初に属性の参照処理について説明する。図30に示すように、出力処理部2330は、ファイル位置情報保持部2310から属性結果(ファイル位置)を1つ取り出し(ステップ2501)、文書ファイルの「読み出し位置」を読み出した照合結果に設定すると共に(ステップ2502)、その読み出し位置に存在する

文書構成要素内から属性を1つ読み込む(ステップ2503)。

【0179】次に出力処理部2330は、その属性名は属性情報指定部2320から指定された属性名と同じであるか否かを判断する(ステップ2504)。

【0180】ここで、同一の場合はその属性値を出力用のファイルに書き出し(ステップ2505)、その後、属性は終りか否かを判断する(ステップ2506)。

【0181】ここで、終りでない場合は、上記ステップ2503に戻りこのステップ以降を実行する。すなわち1つの文書構成要素内に存在する全ての属性についての属性の参照の処理を実施する。

【0182】ステップ2506において属性が終りの場合は、照合結果は終りか、つまりファイル位置情報保持部2310に保持されている全ての照合結果について処理したか否かを判断する(ステップ2507)。

【0183】ここで、未処理の照合結果がある場合には上記ステップ2501に戻りこのステップ以降を実行し、一方、全て処理した場合は属性の参照処理を終了する。

【0184】なおステップ2504において指定された属性名でない場合は何もしないでステップ2506に進む。

【0185】次に属性の変更処理について説明する。図31に示す様に、出力処理部2330は、文書ファイルの「読み出し位置」を先頭に設定すると共に(ステップ2601)、ファイル位置情報保持部2310から照合結果(ファイル位置)を1つ取り出す(ステップ2602)。次に、文書ファイルにおける照合結果の位置までの部分をそのまま出力用のファイルに書き出す(ステップ2603)。

【0186】続いて、文書ファイルの「読み出し位置」を照合結果に設定すると共に(ステップ2604)、その読み出し位置に存在する文書構成要素内から属性を1つ読み込む(ステップ2605)。

【0187】そして、その属性名は属性情報指定部2320から指定された属性名と同じであるか否かを判断する(ステップ2606)。ここで、同一の場合は属性名と指定された属性値とを出力用のファイルに書き出し(ステップ2607)、同一でない場合は、属性名と読み込んだ属性値とを出力用のファイルに書き出す(ステップ2608)。

【0188】ステップ2607あるいはステップ2608を終了したら、属性は終りか否かを判断する(ステップ2609)。ここで、終りでない場合は、上記ステップ2605に戻りこのステップ以降を実行する。すなわち1つの文書構成要素内に存在する全ての属性について属性の変更処理を実施する。

【0189】ステップ2609において属性が終りの場合は、照合結果は終りか、つまりファイル位置情報保持

部2310に保持されている全ての照合結果について処理したか否かを判断する(ステップ2610)。

【0190】ここで、未処理の照合結果がある場合には上記ステップ2602に戻りこのステップ以降を実行し、一方、全て処理した場合は、文書ファイルの最後までをそのまま出力用のファイルに書き出す(ステップ2611)。

【0191】この第5の実施例においては、属性の参照により取り出された属性は、外部の汎用的な演算手段を用いることにより加工することができる。その加工結果を属性情報指定部2320から属性情報として設定することにより、文書内に付加することができる。

【0192】以上説明したように第5の実施例によれば、パターン記述情報と、属性名か属性名及び属性値を設定することにより、構造化文書内のオブジェクト間の関係を利用したパターン、つまり基準となる階層構造のパターンとのパターンマッチングを実施し、マッチした部分(文書構成要素)の属性の参照又は変更操作を実行するようにしたので、属性の参照又は変更の処理を自動化することができる。またこのとき、従来の如く予めスタイルを設定しておく必要がないので、ユーザの作業量を軽減させることができる。

【0193】また特定部分の属性に、他の部分の属性値を元にした演算結果を設定するようなことも可能となる。

【0194】

【発明の効果】以上説明したように、第1の発明によれば、解釈手段によって解釈された文書構造パターンと、再編成手段によって再編成された構造化文書とを照合手段により照合し、更に出力処理手段が、その照合により一致した文書構成要素を前記構造化文書から抽出し出力するようにしているので、構造化文書に対する指定された階層構造パターンに従った文書構成要素の検索処理を行うことができるという利点がある。

【0195】第2の発明によれば、出力処理手段は、照合手段の照合により一致した文書構成要素に対して、指定手段により指定された所定の処理例えば削除、置換、複写などの処理を施した後、出力するようにしているので、構造化文書に対する指定された階層構造パターンに従った文書構成要素に対して、削除、置換、複写などの処理を自動的に実行することができることとなり、構造化文書のバッチ処理が可能となる。

【0196】第3の発明によれば、指定手段により削除処理が指定されると、出力処理手段は、蓄積手段に蓄積されている、照合手段の照合により一致した文書構成要素の親である文書構成要素から、蓄積手段に蓄積されている照合手段の照合により一致した文書構成要素に関する情報を取り除くと共に、当該親である文書構成要素からは削除されない子供の子供の文書構成要素を出力するようにしているので、削除すべき文書構成要素の親の文書構成

要素の内容を自動的に変更することができることとなり、構造化文書に対する編集操作を高率良く行うことができるという利点がある。

【0197】第4の発明によれば、解釈手段が、基準となる文書構成要素同志の接続関係を解釈し、また再編成手段が、格納手段に格納されているファイル内の複数の構造化文書それぞれを照合処理可能な形式の構造に再編成し、また照合手段が、解釈手段による解釈結果と、再編成手段による再編成結果とを照合し、更に出力処理手段が、照合手段の照合により一致した文書構成要素を前記再編成結果から抽出するようにしているので、複数の構造化文書から、指定された階層構造に従った文書構成要素を検索し出力することができるという利点がある。

【0198】第5の発明によれば、複数の構造化文書を保存した複数のファイルを対象として、文書構成要素同志の接続関係に適合する文書構成要素を再編成結果から抽出するようにしているので、複数のファイルそれぞれに複数の構造化文書が存在している場合であっても、複数のファイルおよび複数の構造化文書に跨がって、指定された階層構造に従った文書構成要素を検索し出力することができることとなり、複数の文書に対する検索処理を高速に実行することができるという利点がある。

【0199】第6の発明によれば、ソース側及びターゲット側それぞれの複数の構造化文書を保存した複数のファイルを対象として、文書構成要素同志の接続関係に適合する文書構成要素を再編成結果から抽出すると共に、ターゲット側の抽出結果である文書構成要素に対するソース側の抽出結果である文書構成要素の挿入を実行するようにしているので、ターゲット側の複数のファイルそれぞれに保存されている複数の構造化文書から抽出された文書構成要素に対する、ソース側の複数のファイルそれぞれに保存されている複数の構造化文書から抽出された文書構成要素の挿入を一度に行うことができる。よって文書の編集処理を迅速に行うことができるという利点がある。

【0200】第7の発明によれば、出力処理手段は、指定された属性に関する情報に基づいて、照合手段の照合により一致した文書構成要素の属性の参照又は変更の操作を実行するようにしているので、構造化文書から、指定された階層構造に従った文書構成要素を検索し、この文書構成要素の属性に対する参照又は変更の操作を実施することができることとなり、構造化文書の文書構成要素の属性に対する操作を容易に実施することができるという利点がある。

【図面の簡単な説明】

【図1】本発明に係る文書処理装置の第1の実施例を示す機能ブロック図。

【図2】図1に示した実施例の装置を実現するためのハードウェア構成を示すブロック図。

【図3】第1の実施例における解釈部によるパターン記

述の解釈処理を説明するための図。

【図4】第1の実施例における解釈部の解釈処理動作を示すフローチャート。

【図5】第1の実施例における解釈部によるパターン記述の解釈処理過程を説明するための図。

【図6】第1の実施例における解釈部によるパターン記述の解釈処理過程を説明するための図。

【図7】第1の実施例における再編成部による構造化文書の再編成処理を説明するための図。

【図8】第1の実施例における照合部の照合処理動作を示すフローチャート。

【図9】第1の実施例における照合部による文書構造パターンと再編成構造化文書との照合処理を説明するための図。

【図10】本発明に係る文書処理装置の第2の実施例を示す機能ブロック図。

【図11】第2の実施例における解釈部によるパターン記述の解釈処理を説明するための図。

【図12】第2の実施例における再編成部による構造化文書の再編成処理を説明するための図。

【図13】第2の実施例における照合部による文書構造パターンと再編成構造化文書との照合処理を説明するための図。

【図14】第2の実施例における命令処理部の解釈処理を説明するための図。

【図15】第2の実施例における出力処理部の出力処理を説明するための図。

【図16】本発明に係る文書処理装置の第3の実施例を示す機能ブロック図。

【図17】第3実施例の削除情報の構造の一例を示す図。

【図18】第3実施例における削除情報作成処理を示すフローチャート。

【図19】第3実施例における出力処理部による出力処理動作を示すフローチャート。

【図20】第3実施例における出力処理部による出力処理動作を示すサブルーチン。

【図21】本発明に係る文書処理装置の第4の実施例を示す機能ブロック図。

【図22】第4実施例における構造化文書を説明するための図。

【図23】第4実施例における抽出点認識処理動作を示すフローチャート。

【図24】第4の実施例における照合部による文書構造パターンと再編成構造化文書との照合処理を説明するための図。

【図25】第4の実施例における出力処理部の出力処理動作を示すフローチャート。

【図26】第4の実施例における出力処理部の出力処理を説明するための図。

【図27】第4の実施例における出力処理部の出力処理を説明するための図。

【図28】本発明に係る文書処理装置の第5の実施例を示す機能ブロック図。

【図29】第5実施例における構造化文書を説明するための図。

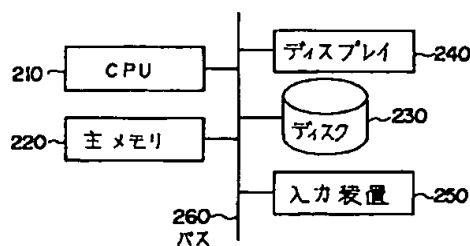
【図30】第5実施例における文書構成要素の属性の参照処理動作を示すフローチャート。

【図31】第5実施例における文書構成要素の属性の変更処理動作を示すフローチャート。

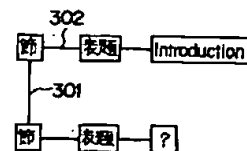
【符号の説明】

10、1610…メモリ、20、1611…パターン記述情報、30、1620…解釈部、40、1612…文書構造パターン、50、1630…再編成部、60…文書ファイル、70…再編成構造化文書、80、1660…照合部、90、1030、1690、2330…出力処理部、210…中央処理装置、220…主メモリ、230…ディスク、240…ディスプレイ、250…入力装置、1010…適合ノード蓄積部、1020…命令処理部、1040…削除情報蓄積部、1613…ソース再編成構造化文書群、1614…ターゲット再編成構造化文書群、1640…ソース文書ファイル群、1650…ターゲット文書ファイル群、1661…複数ファイル抽出点認識部、1662…複数ファイル挿入点認識部、1670…複数抽出点蓄積部、1680…複数挿入点蓄積部、1691…複数ファイル部分構造抽出挿入部、1700…抽出挿入関係ルール指定部、2310…ファイル位置情報保持部、2320…属性情報指定部。

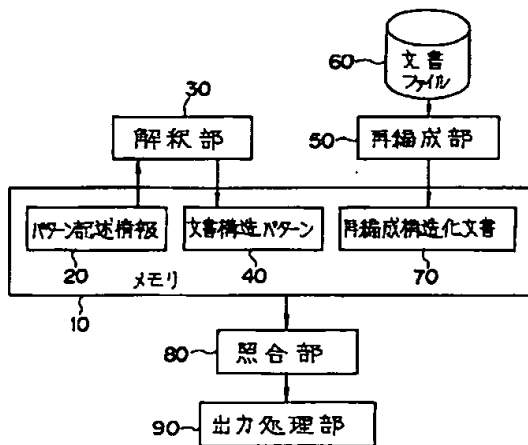
【図2】



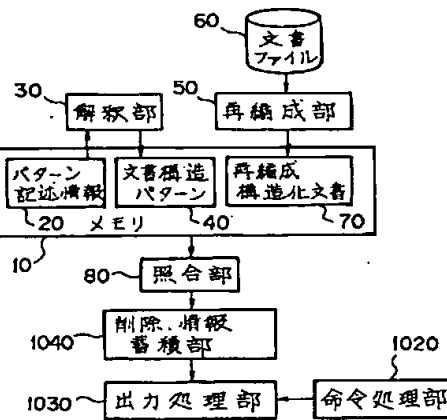
【図3】



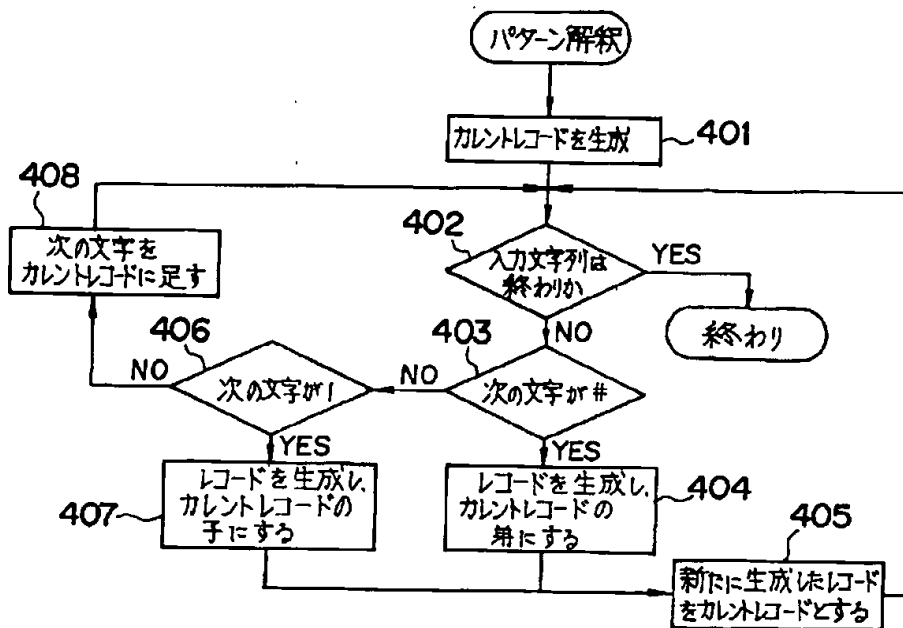
【図1】



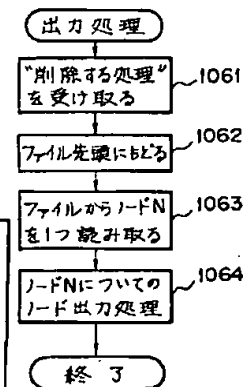
【図16】



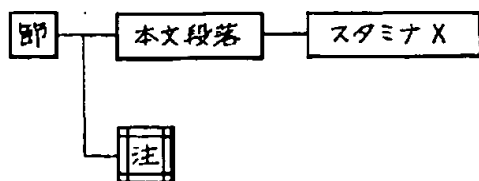
【図4】



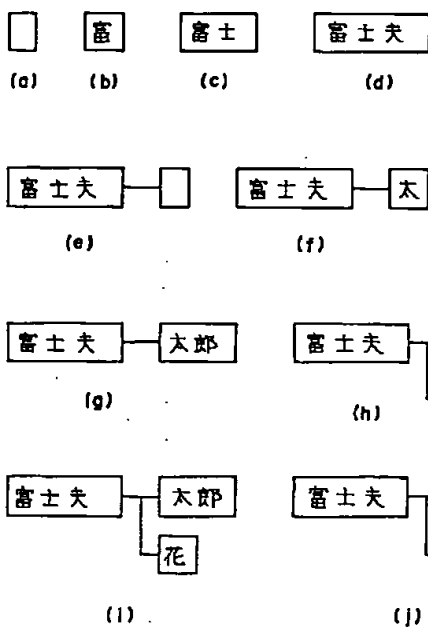
【図19】



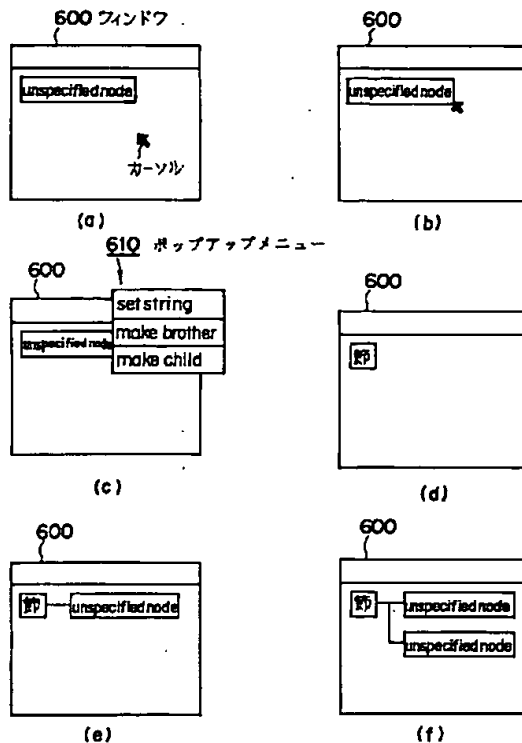
【図11】



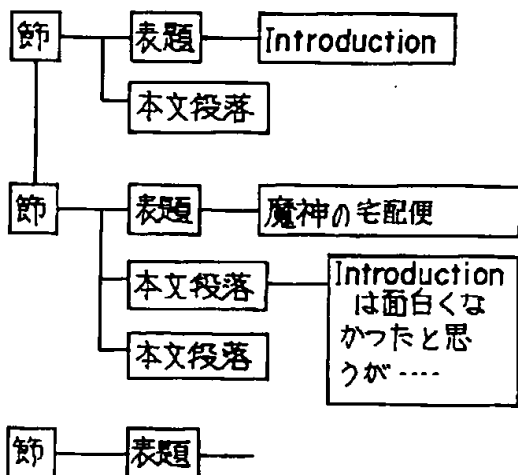
【図5】



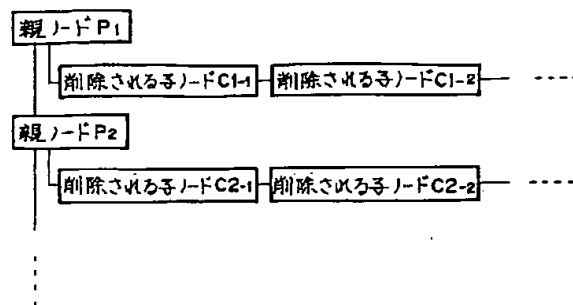
【図6】



【図7】



【図17】



【図8】

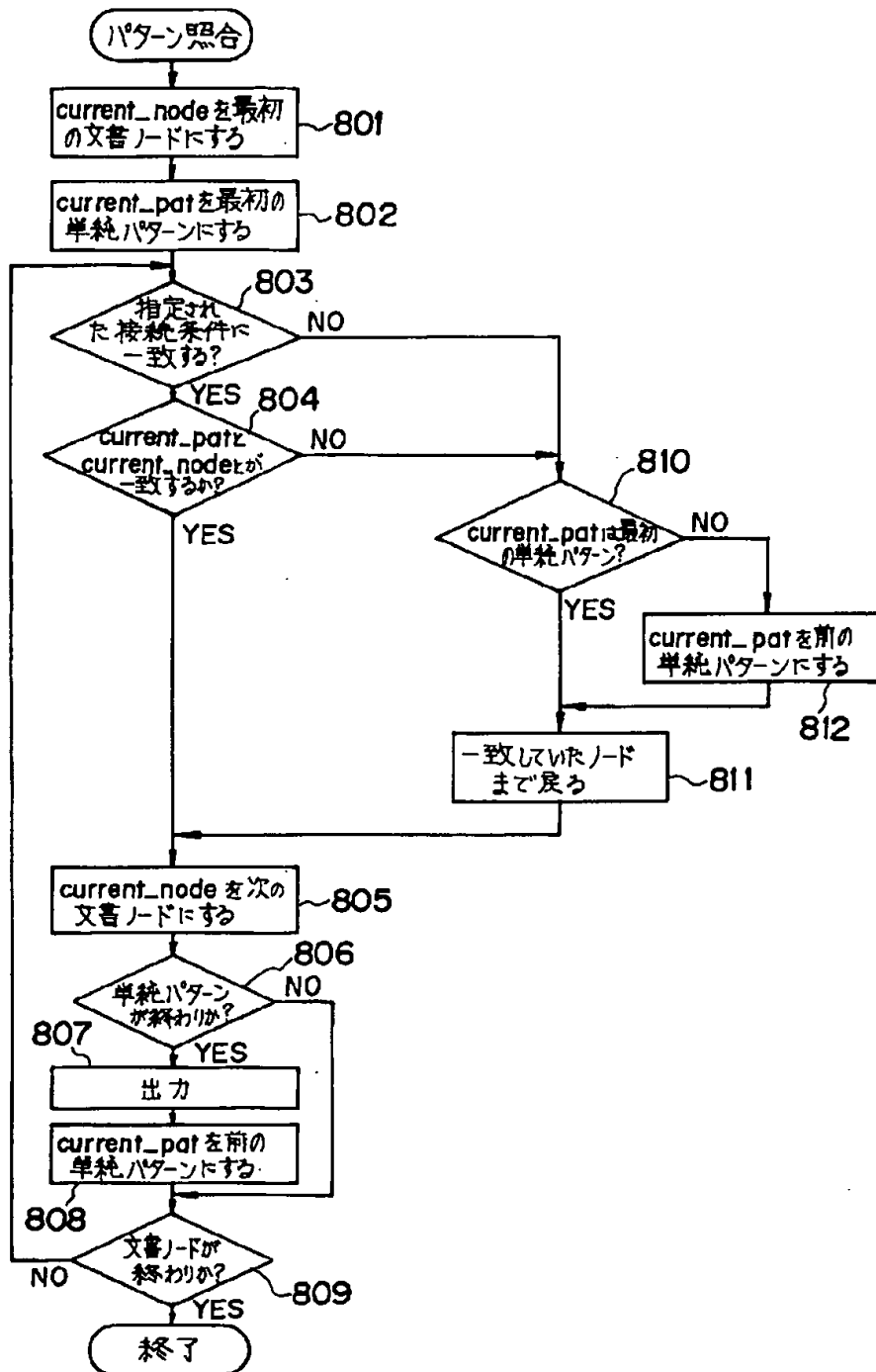


Figure 1 is a hierarchical diagram illustrating the relationship between a document structure and a search engine. The diagram is divided into two main sections: a document structure on the left and a search engine structure on the right.

Document Structure (Left):

- 901A: 節 (Section)
- 902A: 表題 (Title)
- 903A: Introduction
- 904A: 節 (Section)
- 905A: 表題 (Title)
- 906A: ? (Unknown)

Search Engine Structure (Right):

- 901B: 節 (Section)
- 902B: 表題 (Title)
- 903B: Introduction
- 904B: 節 (Section)
- 905B: 表題 (Title)
- 906B: 魔神の宅配便 (Mascot's Mail Delivery)
- 907: Introduction (面白くなかったと思うが ----)

Connections:

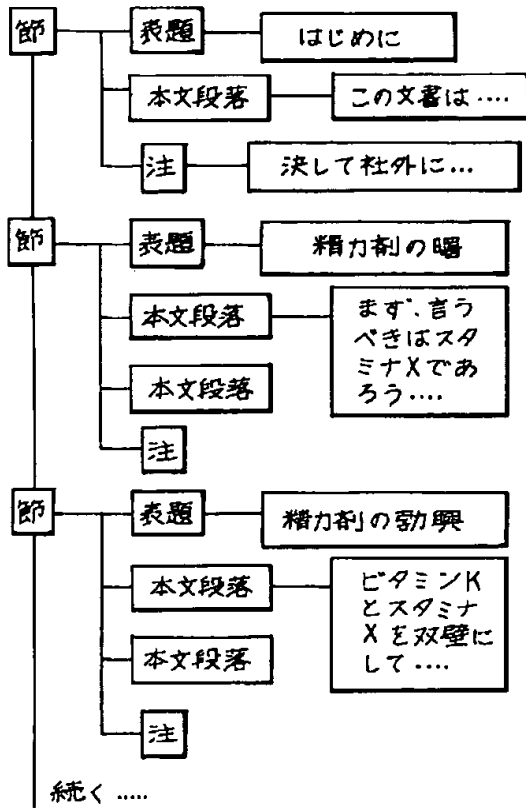
- Solid lines connect the nodes within each structure.
- Dashed lines connect corresponding nodes between the document structure and the search engine structure, indicating a mapping or relationship.

```

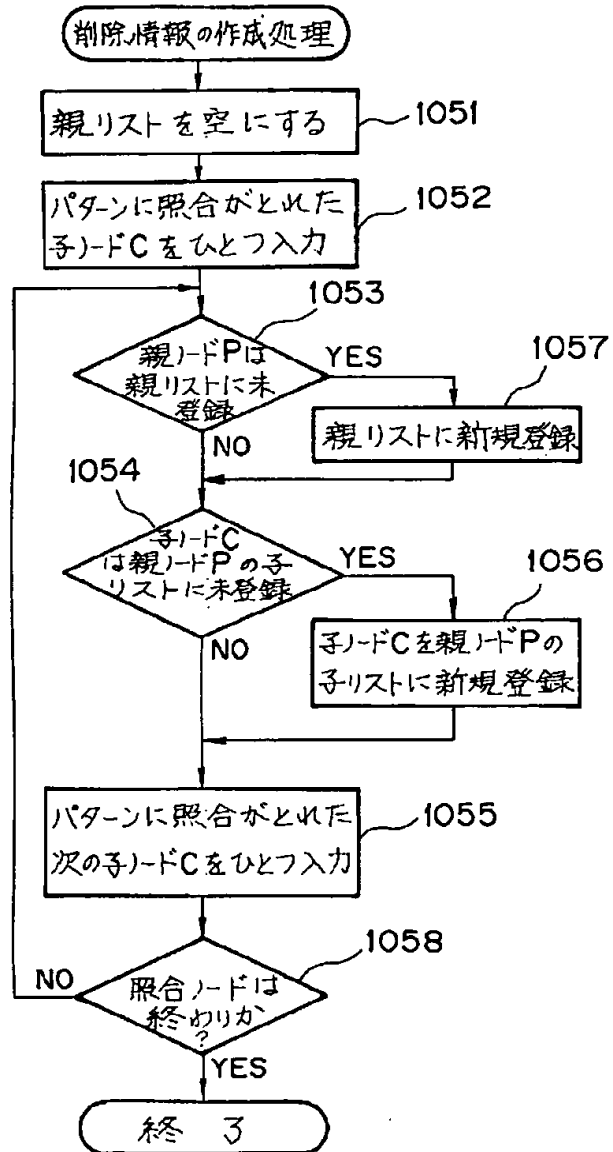
graph TD
    60[(文書ファイル)] --> 50[再編成部]
    50 --> 20[パターン記述情報]
    50 --> 40[文書構造パターン]
    50 --> 70[再編成構造化文書]
    20 --> 30[解析部]
    40 --> 30
    70 --> 30
    30 --> 80[照合部]
    80 --> 1010[適合ノード蓄積部]
    1010 --> 1030[出力処理部]
    1030 --> 1020[命令処理部]
    
```

Figure 1 is a block diagram of the document structure analysis system 10. The system includes a document file 60, a reorganization unit 50, a pattern description information unit 20, a document structure pattern unit 40, a reorganization structured document unit 70, an analysis unit 30, a matching unit 80, a matching node accumulation unit 1010, an output processing unit 1030, and a command processing unit 1020. The document file 60 is connected to the reorganization unit 50. The reorganization unit 50 is connected to the pattern description information unit 20, the document structure pattern unit 40, and the reorganization structured document unit 70. The pattern description information unit 20 is connected to the analysis unit 30. The document structure pattern unit 40 is connected to the analysis unit 30. The reorganization structured document unit 70 is connected to the analysis unit 30. The analysis unit 30 is connected to the matching unit 80. The matching unit 80 is connected to the matching node accumulation unit 1010. The matching node accumulation unit 1010 is connected to the output processing unit 1030. The output processing unit 1030 is connected to the command processing unit 1020.

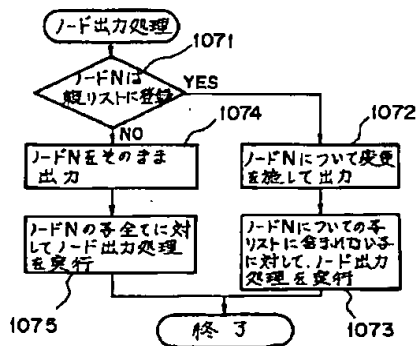
【図12】



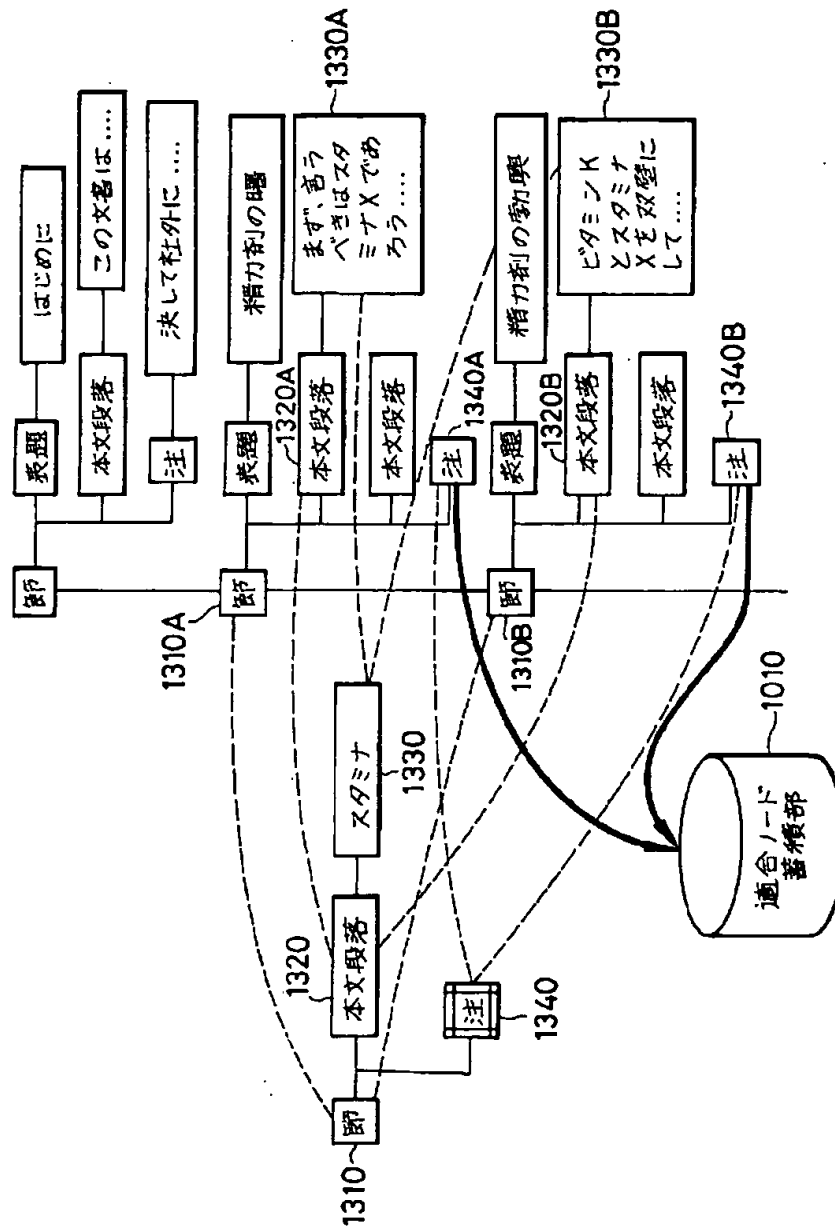
【図18】



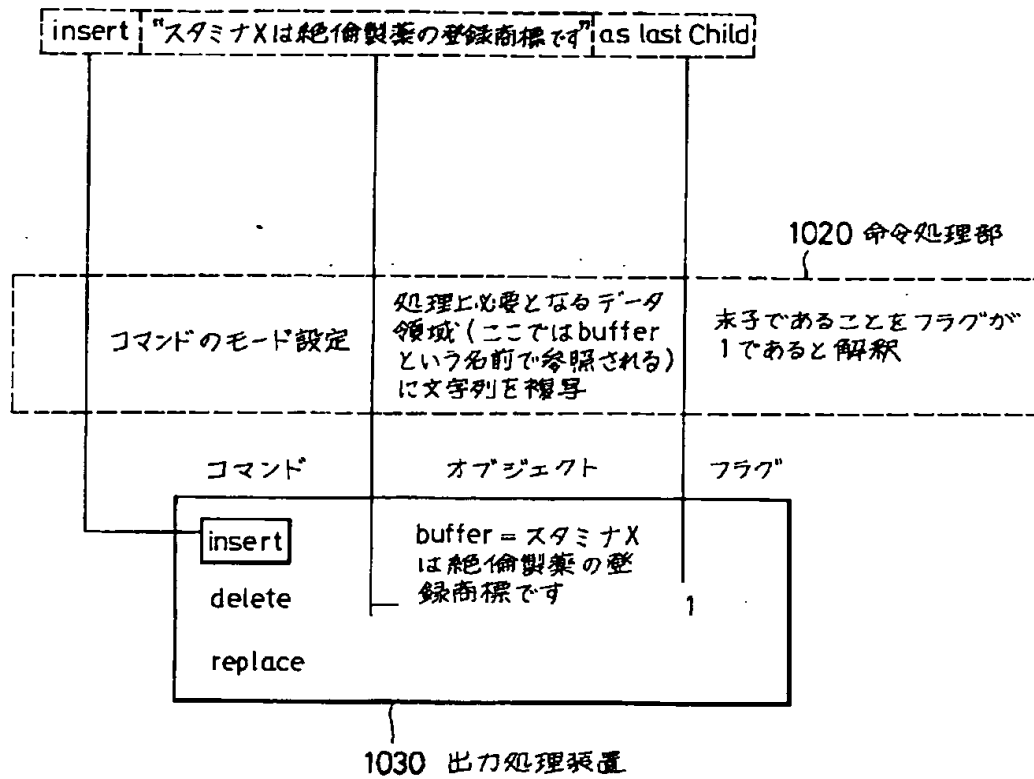
【図20】



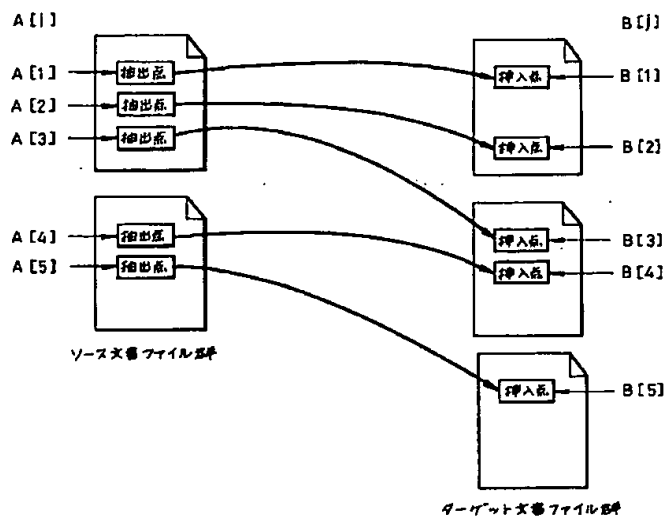
【例 13】



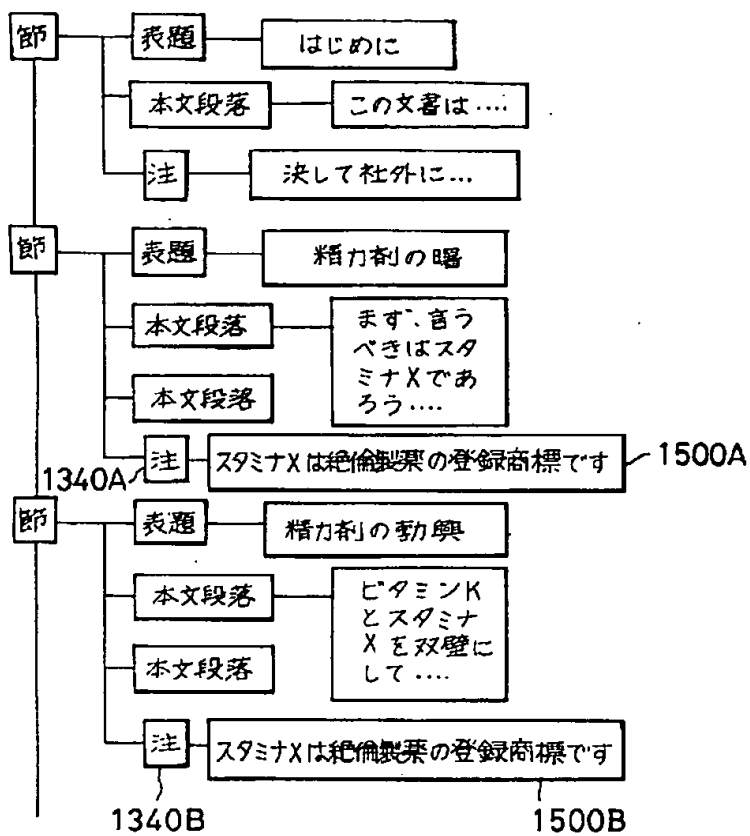
【図14】



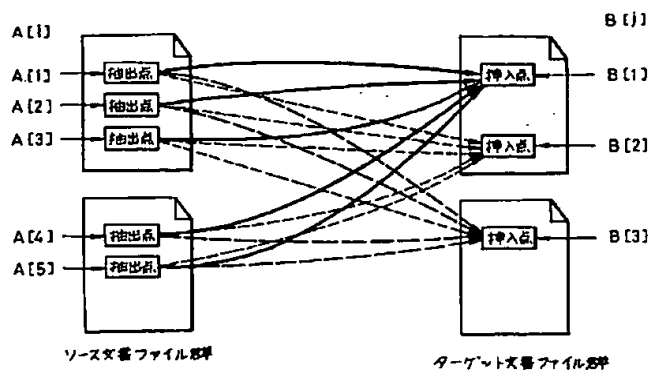
【図26】



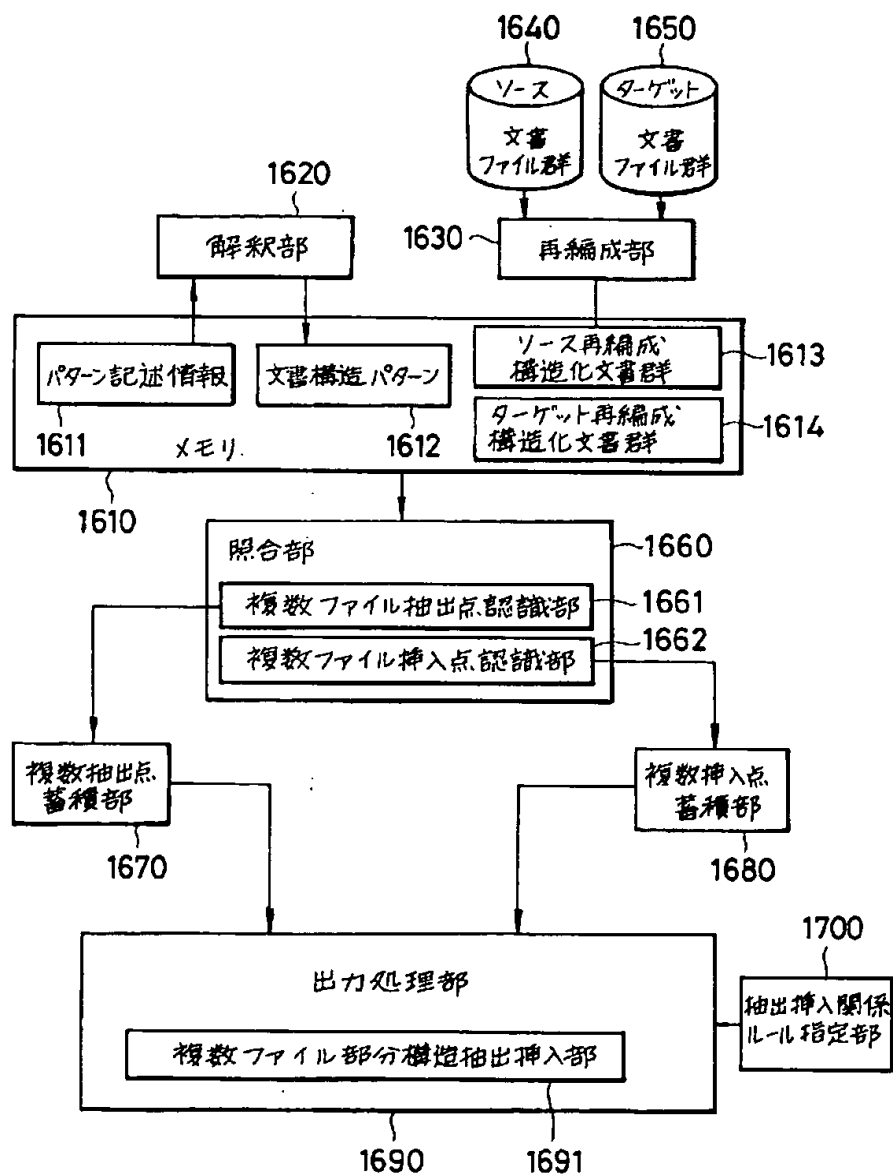
【図15】



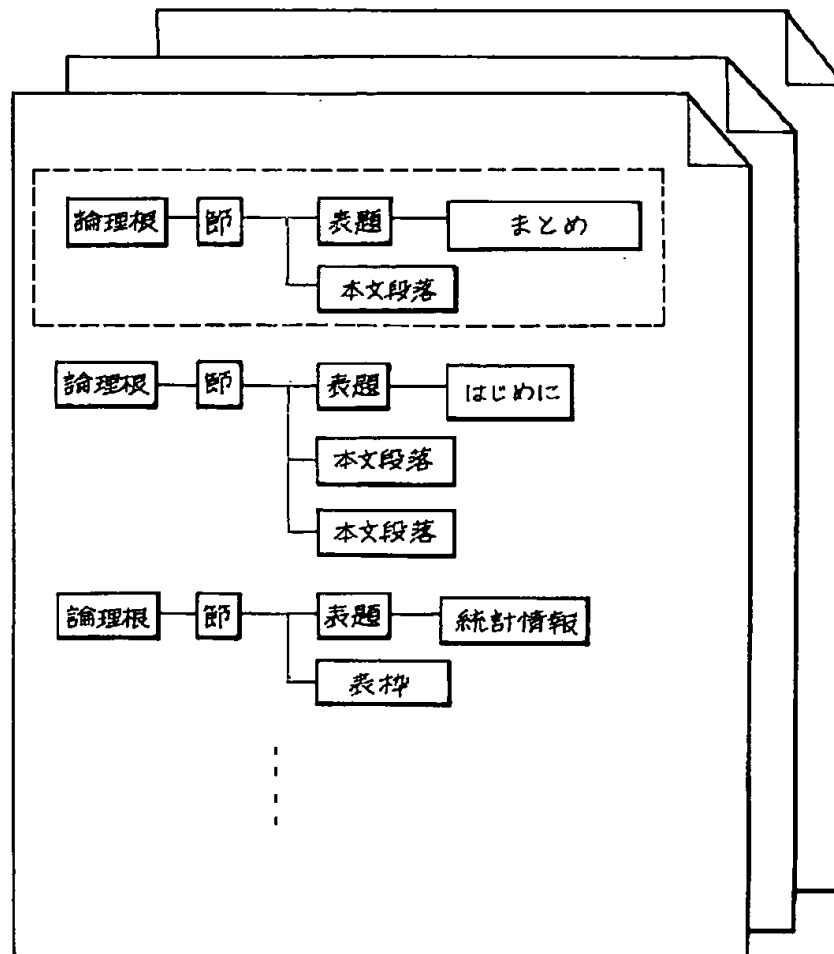
【図27】



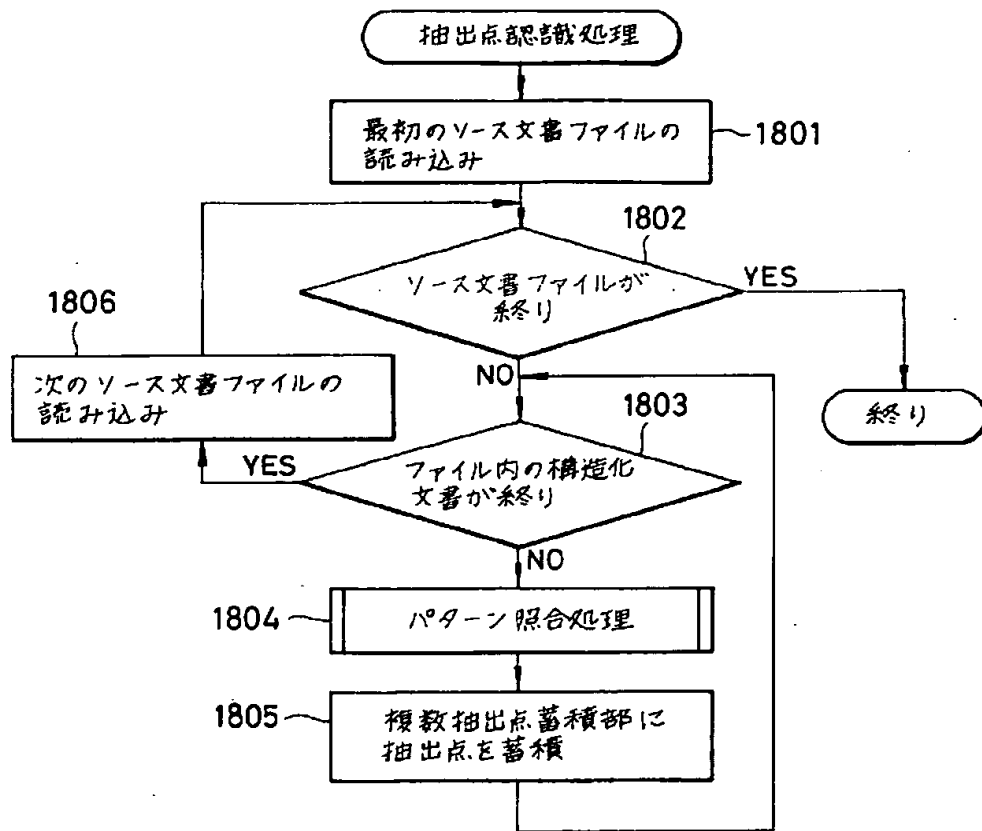
【図21】



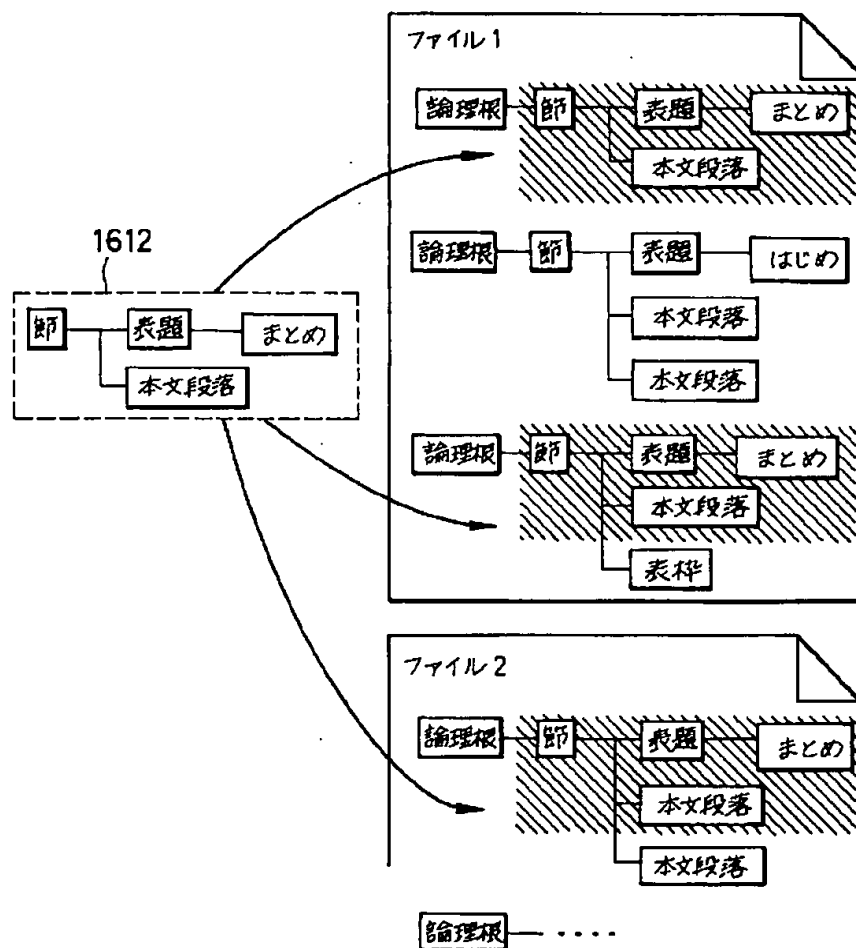
【図22】



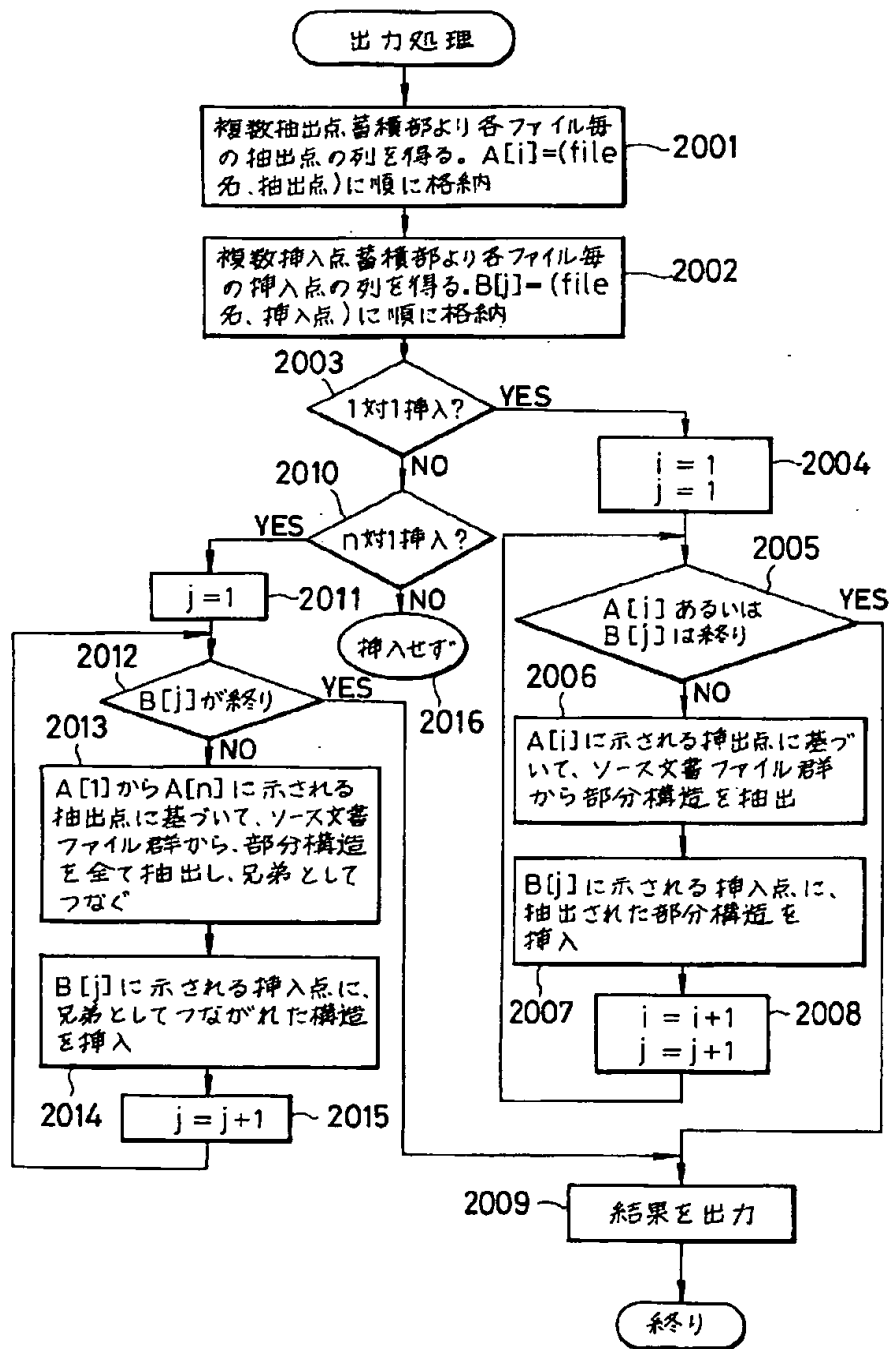
【図23】



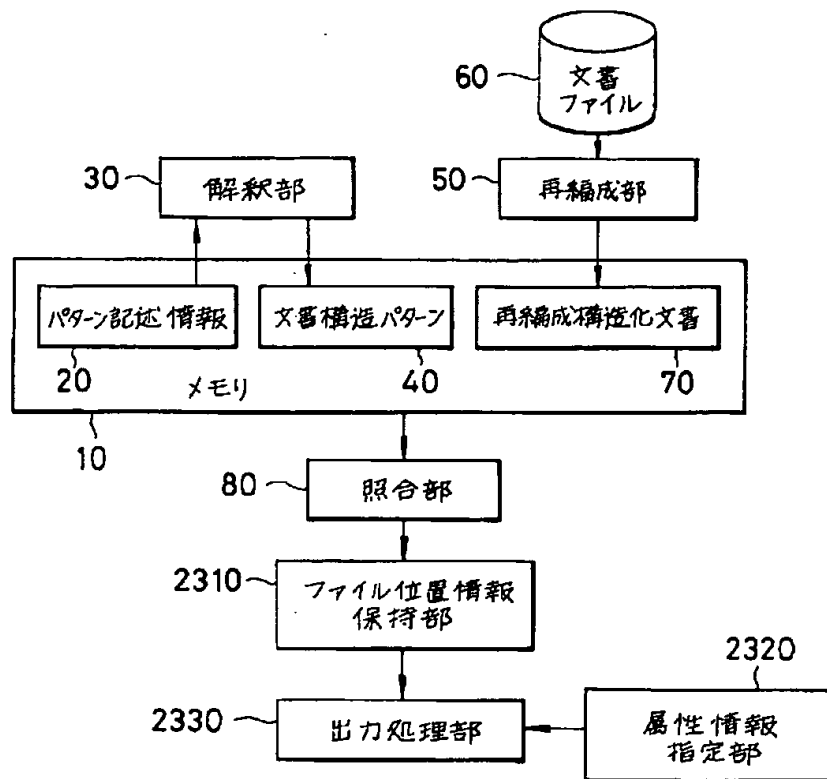
【図24】



【図25】

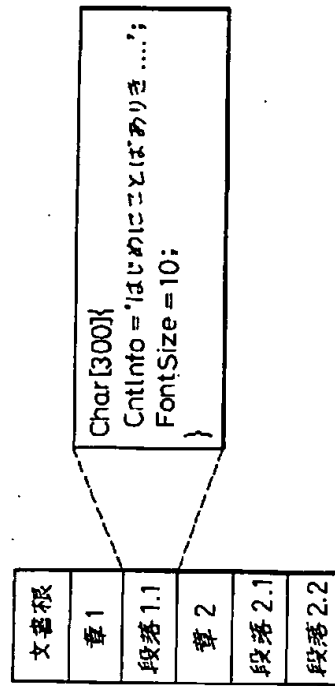


【図28】



【図29】

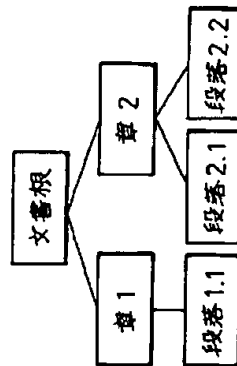
(b)



文書構成要素の表現

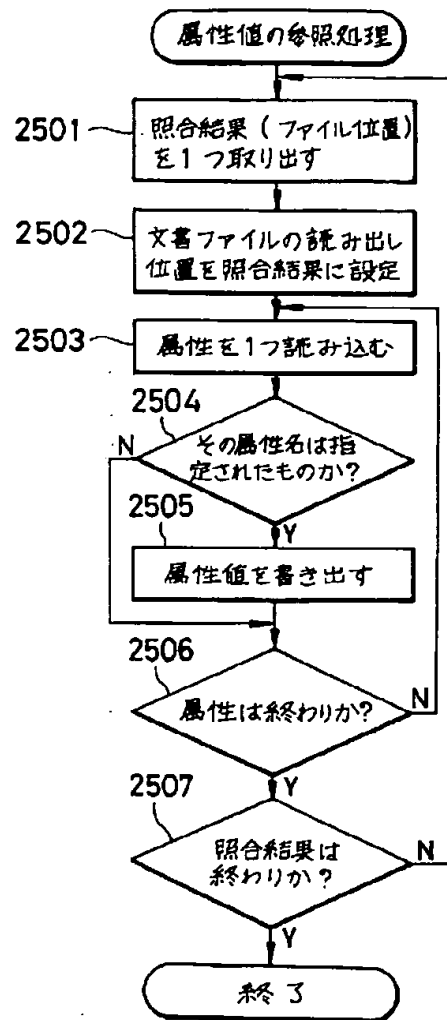
ファイル内の構造

(a)

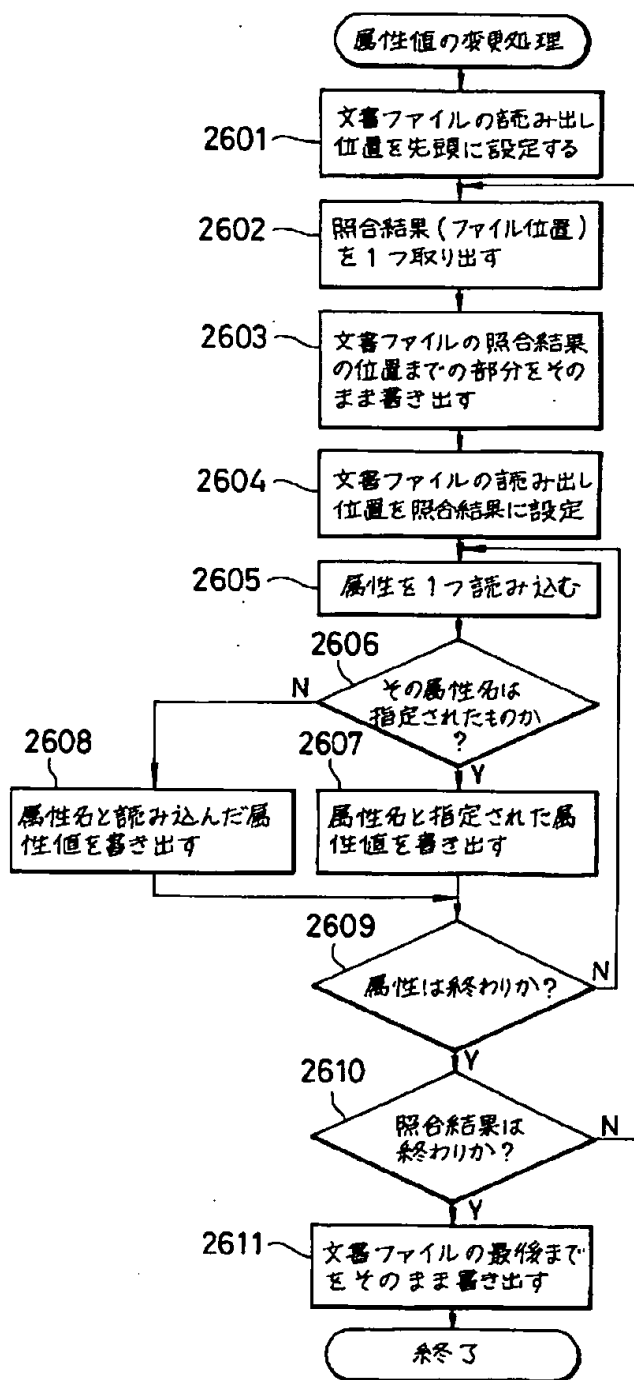


文書の木構造

【図30】



【図31】



フロントページの続き

(72) 発明者 松本 天
神奈川県川崎市高津区坂戸 3 丁目 2 番 1 号
K S P R & D ビジネスパークビル
富士ゼロックス株式会社内

MACHINE-ASSISTED TRANSLATION (MAT):**(19)【発行国】**

日本国特許庁 (J P)

(19)[ISSUING COUNTRY]

Japan Patent Office (JP)

(12)【公報種別】

公開特許公報 (A)

(12)[GAZETTE CATEGORY]

Laid-open Kokai Patent (A)

(11)【公開番号】

特開平 6 - 2 5 9 4 2 1

(11)[KOKAI NUMBER]

Unexamined Japanese Patent Heisei 6-259421

(43)【公開日】平成 6 年 (1 9 9 4) 9 月 1 6
日**(43)[DATE OF FIRST PUBLICATION]**

September 16, Heisei 6 (1994. 9.16)

(54)【発明の名称】

文書処理装置

(54)[TITLE of the Invention]

Word-processing apparatus

(51)【国際特許分類第 5 版】G06F 15/20 550 E
7315-5L
554 M
7315-5L
15/40 500 L
9194-5L**(51)[IPC Int. Cl. 5]**G06F 15/20 550 E 7315-5L
554 M 7315-5L
15/40 500 L 9194-5L**【審査請求】** 未請求**[REQUEST FOR EXAMINATION]** No**【請求項の数】** 7**[NUMBER OF CLAIMS]** 7**【出願形態】** O L**[FORM of APPLICATION]** Electronic**【全頁数】** 3 3**[NUMBER OF PAGES]** 33

(21)【出願番号】

特願平5-164761

(21)[APPLICATION NUMBER]

Japanese Patent Application Heisei 5-164761

(22)【出願日】

平成5年(1993)7月2日

(22)[DATE OF FILING]

July 2, Heisei 5 (1993. 7.2)

(31)【優先権主張番号】

特願平4-176792

(31)[FOREIGN PRIORITY APPLICATION
NUMBER]

Japanese Patent Application Heisei 4-176792

(32)【優先日】

平4(1992)7月3日

(32)[FOREIGN PRIORITY DATE]

July 3, Heisei 4 (1992. 7.3)

(33)【優先権主張国】

日本(JP)

(33)[COUNTRY OF FOREIGN PRIORITY]

(JP)

(31)【優先権主張番号】

特願平5-2855

(31)[FOREIGN PRIORITY APPLICATION
NUMBER]

Japanese Patent Application Heisei 5-2855

(32)【優先日】

平5(1993)1月11日

(32)[FOREIGN PRIORITY DATE]

January 11, Heisei 5 (1993. 1.11)

(33)【優先権主張国】

日本(JP)

(33)[COUNTRY OF FOREIGN PRIORITY]

(JP)

(71)【出願人】

(71)[PATENTEE/ASSIGNEE]

【識別番号】

000005496

[ID CODE]

000005496

【氏名又は名称】

富士ゼロックス株式会社

[NAME OR APPELLATION]

Fuji-Xerox K.K.

【住所又は居所】

東京都港区赤坂三丁目3番5号

[ADDRESS or DOMICILE]

(72) 【発明者】

(72)[INVENTOR]

【氏名】

[NAME OR APPELLATION]

古城 慎太郎

Kojo Shintaro

【住所又は居所】

[ADDRESS or DOMICILE]

神奈川県川崎市高津区坂戸3丁目2番1号 K S P R & D
ビジネスパークビル 富士ゼロックス株式会社内

(72) 【発明者】

(72)[INVENTOR]

【氏名】

[NAME OR APPELLATION]

宮崎 淳

Miyazaki Atsushi

【住所又は居所】

[ADDRESS or DOMICILE]

神奈川県川崎市高津区坂戸3丁目2番1号 K S P R & D
ビジネスパークビル 富士ゼロックス株式会社内

(72) 【発明者】

(72)[INVENTOR]

【氏名】

[NAME OR APPELLATION]

松本 天

Matsumoto Ten

【住所又は居所】

[ADDRESS or DOMICILE]

神奈川県川崎市高津区坂戸3丁目2番1号 K S P R & D
ビジネスパークビル 富士ゼロックス株式会社内

(74) 【代理人】

(74)[AGENT]

【弁理士】

[PATENT ATTORNEY]

【氏名又は名称】

[NAME OR APPELLATION]

木村 高久

Kimura Takahisa

(57) 【要約】

(57)[ABSTRACT of the Disclosure]

【目的】

[PURPOSE]

構造化文書に対する指定された階層構造パターンに従った文書構成要素の検索処理を行うことのできる文書処理装置を提供する。

The word-processing apparatus which can process the search of a document component according to the designated hierarchical-structure pattern with respect to a structuring document is provided.

【構成】

[CONSTITUTION]

メモリ10には、文書構造要素同志の接続関係が記述された情報を示すパターン記述情報20が格納され、また文書ファイル60には構造化文書が格納されている。解釈部30は、パターン記述情報20を解釈して、階層構造を表現している文書構造パターンを生成し、この文書構造パターン40をメモリ10に格納する。再編成部50は、文書ファイル60の構造化文書を走査して、照合処理可能な形式の構造に再編成し、この結果である再編成構造化文書70をメモリ10に格納する。照合部80は文書構造パターン40と再編成構造化文書70とを照合する。出力処理部90は、照合部80の照合により一致したもの

The pattern description information 20 which shows the information a document structure component element's relation of connection was described to be is stored in memory 10, moreover, the structuring document is stored in the text file 60.

The interpretation section 30 interprets the pattern description information 20, the document structure pattern expressing hierarchical structure is generated, this document structure pattern 40 is stored in memory 10.

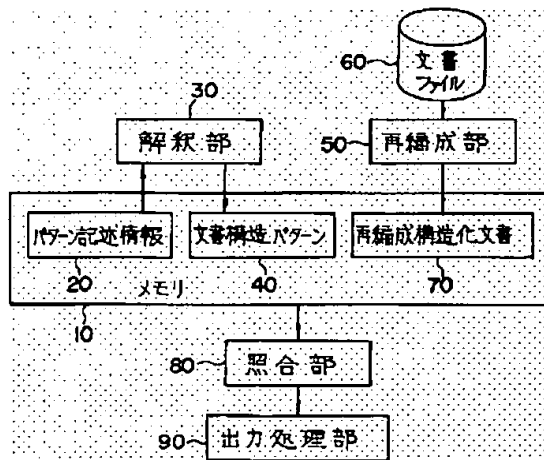
The reorganization section 50 scans the structuring document of a text file 60, it reorganizes in the structure of the form which can carry out collation processing, the reorganization structuring document 70 which is this result is stored in memory 10.

The collation section 80 compares the document structure pattern 40 and the

を出力する。

reorganization structuring document 70.

The output-processing section 90 outputs what was in agreement with collation of the collation section 80.



- 60 Text file
- 30 Interpretation section
- 50 Reorganization section
- 20 Pattern description information
- 40 Document structure pattern
- 70 Reorganization structuring document
- 10 Memory
- 80 Collation section
- 90 Output-processing section

【特許請求の範囲】

[CLAIMS]

【請求項1】

複数の文書構成要素を有する構造化文書に対する処理を行う文書処理装置において、基準となる文書構成要素同志の接続関係を解釈する解釈手段と、

[CLAIM 1]

A word-processing apparatus, in which in the word-processing apparatus which performs processing with respect to a structuring document in which it has two or more document components, interpretation means to interpret the relation of connection of the document

前記構造化文書を照合処理可能な形式の構造に再編成する再編成手段と、
前記解釈手段による解釈結果と、前記再編成手段による再編成結果とを照合する照合手段と、
この照合手段の照合により一致した文書構成要素を前記再編成結果から抽出し出力する出力処理手段とを具備したことを特徴とする文書処理装置。

【請求項 2】

所定の文書構成要素に対する所定の処理を指定する指定手段を更に具備し、
前記出力処理手段は、前記照合手段の照合により一致した文書構成要素に対して、前記指定手段により指定された所定の処理を施した後、出力することを特徴とする請求項 1 記載の文書処理装置。

【請求項 3】

所定の文書構成要素に対する削除処理を指定する指定手段と、
前記照合手段の照合により一致した文書構成要素と、当該文書構成要素の親である文書構成要素とを対応付けして蓄積する蓄積手段とを更に具備し、
前記出力処理手段は、前記指定手段により削除処理が指定されると、前記蓄積手段に蓄積され

component element who becomes a reference standard, reorganization means to reorganize said structuring document in the structure of the form which can carry out collation processing, collation means to compare the interpretation result by said interpretation means, and the reorganization result by said reorganization means, output-processing means to have extracted the document component which was in agreement with collation of this collation means from said reorganization result, and to output it were comprised.

[CLAIM 2]

A word-processing apparatus of Claim 1, in which designation means to designate the fixed processing with respect to a fixed document component are comprised further, said output-processing means are outputted after performing fixed processing designated by said designation means to the document component which was in agreement with collation of said collation means.

[CLAIM 3]

A word-processing apparatus of Claim 1, in which designation means to designate the deletion processing with respect to a fixed document component, and accumulation means to match and accumulate the document component which was in agreement with collation of said collation means, and the document component which is the parents of said document component are comprised further.

ている前記親である文書構成要素から、前記蓄積手段に蓄積されている前記照合手段の照合により一致した文書構成要素に関する情報を取り除くと共に、当該親である文書構成要素から削除されない、当該親の文書構成要素の子供である文書構成要素を出力することを特徴とする請求項1記載の文書処理装置。

Said output-processing means, if deletion processing is designated by said designation means, while removing the information about the document component which was in agreement with collation of said collation means accumulated in said accumulation means from the document component which is said parents accumulated in said accumulation means, the document component which is not deleted from the document component which is said parents and which is the child of said parents' document component is outputted.

【請求項4】

複数の文書構成要素を有する構造化文書を複数保存したファイルを格納する格納手段と、
基準となる文書構成要素同志の接続関係を解釈する解釈手段と、
前記格納手段に格納されているファイル内の複数の構造化文書それぞれを照合処理可能な形式の構造に再編成する再編成手段と、
前記解釈手段による解釈結果と、前記再編成手段による再編成結果とを照合する照合手段と、
この照合手段の照合により一致した文書構成要素を前記再編成結果から抽出し出力する出力処理手段とを具備したことを特徴とする文書処理装置。

[CLAIM 4]

A word-processing apparatus, in which storing means to store the file which conserved two or more structuring documents which have two or more document components, interpretation means to interpret the relation of connection of the document component element who becomes a reference standard, reorganization means to reorganize two or more structuring documents of each in the file stored in said storing means in the structure of the form which can carry out collation processing, collation means to compare the interpretation result by said interpretation means, and the reorganization result by said reorganization means, output-processing means to have extracted the document component which was in agreement with collation of this collation means from said reorganization result, and to output it were comprised.

【請求項5】**[CLAIM 5]**

前記複数の構造化文書を保存した複数のファイルを対象として、前記文書構成要素同志の接続関係に適合する文書構成要素を前記再編成結果から抽出することを特徴とする請求項4記載の文書処理装置。

【請求項6】

ソース側及びターゲット側それぞれの前記複数の構造化文書を保存した複数のファイルを対象として、前記文書構成要素同志の接続関係に適合する文書構成要素を前記再編成結果から抽出すると共に、前記ターゲット側の抽出結果である文書構成要素に対する前記ソース側の抽出結果である文書構成要素の挿入を実行することを特徴とする請求項4記載の文書処理装置。

【請求項7】

前記出力処理手段は、指定された属性に関する情報に基づいて、前記照合手段の照合により一致した文書構成要素の属性の参照又は変更の操作を実行することを特徴とする請求項2乃至請求項6記載の文書処理装置。

【発明の詳細な説明】

A word-processing apparatus of Claim 4, in which the document component which adapts said document component element's relation of connection is extracted from said reorganization result by making into object two or more files which conserved these structuring documents.

[CLAIM 6]

A word-processing apparatus of Claim 4, in which while extracting the document component which adapts the above-mentioned document component element's connection concern for two or more files which conserved each of these structuring document by the side of a source and a target from the above-mentioned reorganization result, insertion of the document component which is an extraction result by the side of said source with respect to the document component which is the extraction result of said target- is performed.

[CLAIM 7]

A word-processing apparatus of claims 2 - 6, in which said output-processing means perform the operation of refer or alteration of the attribute of a document component which was in agreement with collation of said collation means based on the information about the designated attribute.

[DETAILED DESCRIPTION of the INVENTION]

【0001】**【産業上の利用分野】**

この発明は、文書中から所望のパターンを検索する文書処理装置に関する。

[0001]**[INDUSTRIAL APPLICATION]**

This invention relates to the word-processing apparatus with which a desired pattern is searched out of a document.

【0002】**【従来の技術】**

従来においては、ワードプロセッサや、文書作成・編集機能を有するワークステーションやコンピュータ等の装置には、一般的に、作成された文書中から所望の文字列を検索することのできる検索機能が設けられている。この検索機能を利用することにより、検索対象となる文字列を、他の文字列に置換したり削除したりして、文書を編集することができる。

[0002]**[PRIOR ART]**

In the past, the search service which can search a desired character string out of the document drawn up generally is provided in apparatus, such as a word processor, the workstation which has a documentation * editing function, and a computer.

By utilizing this search service, the character string used as search object is replaced by other character strings, or is deleted, a document can be edited.

【0003】

またこのような文書編集を自動化するようにしたものも実現されており、例えば、

(1) カット・アンド・ペーストしながらの操作をマクロ記述して何度でも実行することができるようにしたもの。

(2) 例えば特開平3-147062号公報に開示されている様に、文書中から取り出された複数の文字列を一時記憶領域に保存し、その後順次、ペースト

[0003]

Moreover, what automated such a text editing is implemented, for example, (1) The thing carries out the macro description of operation while doing the cut * and * paste of, and it enabled it to perform any number of times.

(2) For example, the thing which conserves two or more character strings taken out of the document at temporary storage, and was made to paste (insertion) them in order after that as disclosed by Unexamined-Japanese-Patent No. 3-147062.

(3) What was made to implement replacement

(挿入) するようにしたもの。

(3) 指定したパターン文字列が文字列中に現れた際に、置き換え等を実施する様にしたもの(例えばUNIXのsedのようなストリームエディタ)。がある。

etc. when the designated pattern character string appeared in the character string (for example, a stream editor like UNIX sed).

【0004】

また文書ファイルを編集する方法としては、インタラクティブにユーザがカット・アンド・ペーストする方法や、バッチ処理で行う方法がある。このうち、効率良く複数回の抽出挿入を行う方法として、例えば、

(4) ユーザが指定した抽出文字列を複数個同時に記憶しておき、これら複数の文字列を順次、所定の位置に挿入するようにしたもの(例えば特開平3-147062号公報)。

(5) ユーザがインタラクティブに文書にマークを付与することにより、一度に複数の指定領域の入れ替えを行うようにしたもの(例えば特開平3-260761号公報)。

(6) ファイル内のある特定の文字列パターンの出現に対して、文字列の置き換え等の処理を行うようにしたもの(例えばUNIXのawk、sedなど)。

(7) 構造と内容とを分離して管理する構造化文書システムに

[0004]

Moreover, as a method of editing a text file, there are a method which a user makes interactive in a cut * and * paste, and the method of performing by a batch processing.

Among these, as a method of performing efficient several times of extraction insertion, for example, (4) What stores simultaneously two or more extraction character strings specified by a user, and inserted the character string of these plurality in the position in order (for example, Unexamined-Japanese-Patent No. 3-147062), (5) What was made to replace two or more appointed region at once when a user provided a mark in a document interactively (for example, Unexamined-Japanese-Patent No. 3-260761), (6) What was made to process replacement of a character string etc. to the advent of a certain specific character string pattern in a file (for example, UNIX awk, sed etc.), (7) Thing in which extract the insertion character string beforehand and made it make this insertion character string reflect from two or more documents with insertion designation to an insertion object document in the structuring document system which separates and manages the structure and the content (for example, Unexamined-Japanese-Patent No.

において、差し込み指定のある複数の文書から、差し込み文字列を予め抽出しておき、この差し込み文字列を差し込み対象文書へ反映させるようにしたもの（例えば特開平4-57151号公報）。がある。

【0005】

更に文書の特定部分の属性値（例えば文字の大きさなど）を変更するようにしたものも実現されており、この方法として、例えば、

（8）インタラクティブなエディタにより逐一変更するようにしたもの。

（9）特定部分にスタイルを予め設定しておき、スタイルの属性値を変更することにより、一括変換するようにしたもの。がある。

[0005]

Furthermore, what altered the attribute value (for example, size of a character etc.) of the specific part of a document is implemented, as this method for example, (8) What was altered one-by-one with the interactive editor.

(9) What was made to carry out a package conversion by setting the style as the specific part beforehand and altering the attribute value of a style.

【0006】

【発明が解決しようとする課題】

しかしながら、上記従来の装置では、文書中から文字パターン（検索対象の文字列）を検索することは可能であるが、これは文書中の文字列と指定文字列との照合により一致したものを検索するようにしているので、内部的に階層構造の情報を持った文書いわゆる構造化文書から、

[0006]

[PROBLEM to be solved by the Invention]

However, although a character pattern (character string of search object) can be searched with the above-mentioned conventional apparatus out of a document, this searches what was in agreement with collation with the character string in a document, and a designation character string.

Therefore, the document component which followed the designated hierarchical-structure pattern from the document which had the

指定された階層構造パターンに従った文書要素を検索することができなかった。

【0007】

すなわち、構造化文書においては、文書要素が階層構造上のどこに位置するかによって、その文書要素の情報は意味を変えるので、従来の如く、単に文字列の照合のみにより検索し、階層構造を考慮しない検索では、適切な部分にアクセスすることができず、正確な検索処理を行うことができなかった。

【0008】

また上記(1)のものでは、不正確な処理を施してしまうことが多く、このため、複数の文書をバッチで処理することができない。

【0009】

また上記(2)のものでは、正確な処理を施すことはできるが、バッチで処理することができない。

【0010】

また上記(3)のものでは、バッチ処理に適しているが、構造を持った文書の文字列は構造上の位置によって意味を変えるので、不適切な部分を書き換えてしまう恐れがあり、構造化文書

information on hierarchical structure internally, and the so-called structuring document was not able to be searched.

[0007]

That is, in a structuring document, a document component is positioned where on hierarchical structure, or the information on the document component changes the meaning.

Therefore, in the search which searches only by collation of a character string and does not consider hierarchical structure in accordance with the prior art, it cannot access into a suitable part, exact search processing was not able to be performed.

[0008]

Moreover, in said (1), inaccurate processing is performed in many cases and, for this reason, two or more documents cannot be processed in batch.

[0009]

Moreover, although exact processing can be performed in said (2), it cannot process in batch.

[0010]

Moreover, in said (3), it is suitable for the batch processing.

However, the character string of a document with the structure changes the meaning with the position on the structure.

Therefore, there is a possibility of overwriting an

には適していない。

unsuitable part.

It is not suitable for a structuring document.

【0011】

また上記（４）のものでは、文書中の全指定箇所をユーザがインタラクティブに一度設定しなければならないので、操作が複雑である。

[0011]

Moreover, in said (4), since a user has to set up all the appointed places in a document once interactively, operation is complicated.

【0012】

また上記（５）のものでは、上記（４）の場合と同様に、文書中の全指定箇所をユーザがインタラクティブに一度設定しなければならないので、操作が複雑である。

[0012]

Moreover, in said (5), like the case of said (4), since a user has to set up all the appointed places in a document once interactively, operation is complicated.

【0013】

また上記（６）のものでは、ユーザが文書中の所定箇所を指定する必要はないが、一般的なストリングマッチのため、ファイル中の構造の意味を解釈せずに、単に文字列として照合し、一致した文字列に対して変更を加えるようにしているので、文書の構造の意味を維持しつつ処理することはできない。

[0013]

Moreover, in said (6), a user does not need to designate the fixed location in a document. However, for a general string match, without interpreting the meaning of the structure in a file, it only compares as a character string and alteration is added to the congruous character strings. Therefore, it cannot process, maintaining the meaning of the structure of a document.

【0014】

また上記（７）のものでは、内容の変更については効率良く実行することができるが、構造と内容を同時に変更することはできない。

[0014]

Moreover, in said (7), it can perform efficiently about alteration of the content. However, the structure and the content cannot be altered simultaneously.

【0015】

また上記（８）の方法では、インタラクティブに処理するので、属性値の変更処理を自動化することができず、このため効率が悪く、操作ミスによる間違いも発生し易い。

[0015]

Moreover, by the method of said (8), it processes interactively. Therefore, alteration processing of attribute value is not automatable, and for this reason, effectiveness is bad and also tends to produce the mistake by a failure.

【0016】

更に上記（９）のものでは、予めスタイルを設定しておく必要があるので、そのための手間がかかる。また同一のスタイルには全て同一の属性が付与されるので、柔軟性に欠ける。

[0016]

Furthermore, in said (9), it is the need of setting up the style beforehand. Therefore, it takes time and effort for it. Moreover, the same style provides the same attribute altogether. Therefore, flexibility is missing.

【0017】

そこで本発明の目的は、構造化文書に対する指定された階層構造パターンに従った文書構成要素の検索処理を行うことができる文書処理装置を提供することである。

[0017]

Then, objective of the invention is providing the word-processing apparatus which can process the search of a document component according to the designated hierarchical-structure pattern with respect to a structuring document.

【0018】

本発明の他の目的は、構造化文書に対する指定された階層構造パターンに従った文書構成要素を検索し、この文書構成要素に対する削除、置換、複写などの操作を実施することができる文書処理装置を提供することである。

[0018]

Other objectives of this invention are providing the word-processing apparatus which can search the document component according to the designated hierarchical-structure pattern with respect to a structuring document, and can operate deletion, displacement, reproduction, etc. with respect to this document component.

【0019】**[0019]**

本発明の他の目的は、複数の構造化文書に対する指定された階層構造パターンに従った文書構成要素の検索処理を行うことができる文書処理装置を提供することである。

【0020】

本発明の他の目的は、ターゲット側の複数のファイルそれぞれに保存されている複数の構造化文書から抽出された文書構成要素に対する、ソース側の複数のファイルそれぞれに保存されている複数の構造化文書から抽出された複数の文書構成要素の挿入操作を一度に自動的に行うことができる文書処理装置を提供することである。

【0021】

本発明の他の目的は、複数の構造化文書に対する指定された階層構造パターンに従った文書構成要素を検索し、この文書構成要素の属性値の参照又は変更を行うことができる文書処理装置を提供することである。

【0022】**【課題を解決するための手段】**

第1の発明は、複数の文書構成要素を有する構造化文書に対する処理を行う文書処理装置において、基準となる文書構成要素

Other objectives of this invention are providing the word-processing apparatus which can process the search of a document component according to the designated hierarchical-structure pattern with respect to two or more structuring documents.

[0020]

Other objectives of this invention are providing the word-processing apparatus with respect to the document component extracted from two or more structuring documents conserved at two or more files of each by the side of a target which can perform automatically at once insertion operation of two or more document components extracted from two or more structuring documents conserved at two or more files of each by the side of a source.

[0021]

Other objectives of this invention are providing the word-processing apparatus which can search the document component according to the designated hierarchical-structure pattern with respect to two or more structuring documents, and can make refer or alteration of the attribute value of this document component.

[0022]**[MEANS to solve the Problem]**

1st invention, in the word-processing apparatus which performs processing with respect to a structuring document in which it has two or more document components, interpretation

同志の接続関係を解釈する解釈手段と、前記構造化文書を照合処理可能な形式の構造に再編成する再編成手段と、前記解釈手段による解釈結果と、前記再編成手段による再編成結果とを照合する照合手段と、この照合手段の照合により一致した文書構成要素を前記再編成結果から抽出し出力する出力処理手段とを具備している。

means to interpret the relation of connection of the document component element who becomes a reference standard, reorganization means to reorganize said structuring document in the structure of the form which can carry out collation processing, collation means to compare the interpretation result by said interpretation means, and the reorganization result by said reorganization means, output-processing means to extract the document component which was in agreement with collation of this collation means from said reorganization result, and to output it are comprised.

【0023】

第2の発明は、第1の発明において、所定の文書構成要素に対する所定の処理を指定する指定手段を更に具備し、前記出力処理手段は、前記照合手段の照合により一致した文書構成要素に対して、前記指定手段により指定された所定の処理を施した後、出力することを特徴としている。

[0023]

2nd invention, in 1st invention, designation means to designate the fixed processing with respect to a fixed document component are comprised further, said output-processing means are outputted after performing fixed processing designated by said designation means to the document component which was in agreement with collation of said collation means.

It is characterized by the above-mentioned.

【0024】

第3の発明は、第1の発明において、所定の文書構成要素に対する削除処理を指定する指定手段と、前記照合手段の照合により一致した文書構成要素と、当該文書構成要素の親である文書構成要素とを対応付けして蓄積する蓄積手段とを更に具備し、

[0024]

3rd invention is set in 1st invention, designation means to designate the deletion processing with respect to a fixed document component in 1st invention, and accumulation means to match and accumulate the document component which was in agreement with collation of said collation means, and the document component which is the parents of

前記出力処理手段は、前記指定手段により削除処理が指定されると、前記蓄積手段に蓄積されている前記親である文書構成要素から、前記蓄積手段に蓄積されている前記照合手段の照合により一致した文書構成要素に関する情報を取り除くと共に、当該親である文書構成要素から削除されない、当該親の文書構成要素の子供である文書構成要素を出力することを特徴としている。

said document component are comprised further.

Said output-processing means, if deletion processing is designated by said designation means, while removing the information about the document component which was in agreement with collation of said collation means accumulated in said accumulation means from the document component which is said parents accumulated in said accumulation means, the document component which is not deleted from the document component which is said parents and which is the child of said parents' document component is outputted.

It is characterized by the above-mentioned.

【0025】

第4の発明は、複数の文書構成要素を有する構造化文書を複数保存したファイルを格納する格納手段と、基準となる文書構成要素同志の接続関係を解釈する解釈手段と、前記格納手段に格納されているファイル内の複数の構造化文書それぞれを照合処理可能な形式の構造に再編成する再編成手段と、前記解釈手段による解釈結果と、前記再編成手段による再編成結果とを照合する照合手段と、この照合手段の照合により一致した文書構成要素を前記再編成結果から抽出し出力する出力処理手段とを具備している。

[0025]

4th invention, storing means to store the file which conserved two or more structuring documents which have two or more document components, interpretation means to interpret the relation of connection of the document component element who becomes a reference standard, reorganization means to reorganize two or more structuring documents of each in the file stored in said storing means in the structure of the form which can carry out collation processing, collation means to compare the interpretation result by said interpretation means, and the reorganization result by said reorganization means, and output-processing means to extract the document component which was in agreement with collation of this collation means from said reorganization result, and to output it are comprised.

【0026】

第5の発明は、第4の発明において、前記複数の構造化文書を保存した複数のファイルを対象として、前記文書構成要素同志の接続関係に適合する文書構成要素を前記再編成結果から抽出することを特徴としている。

[0026]

5th invention, in 4th invention, it is characterized by extracting the document component which adapts said document component element's relation of connection from said reorganization result by making into object two or more files which conserved these structuring documents.

【0027】

第6の発明は、第4の発明において、ソース側及びターゲット側それぞれの前記複数の構造化文書を保存した複数のファイルを対象として、前記文書構成要素同志の接続関係に適合する文書構成要素を前記再編成結果から抽出すると共に、前記ターゲット側の抽出結果である文書構成要素に対する前記ソース側の抽出結果である文書構成要素の挿入を実行することを特徴としている。

[0027]

6th invention, while extracting the document component which adapts said document component element's relation of connection in 4th invention by making into object two or more files as for which the source and target side conserved each of these structuring document from said reorganization result, insertion of the document component which is an extraction result by the side of said source with respect to the document component which is an extraction result by the side of said target is performed. It is characterized by the above-mentioned.

【0028】

第7の発明は、第2の発明乃至第6の発明において、前記出力処理手段は、指定された属性に関する情報に基づいて、前記照合手段の照合により一致した文書構成要素の属性の参照又は変更の操作を実行することを特徴としている。

[0028]

7th invention, in 2nd invention or 6th invention, said output-processing means perform the operation of refer or alteration of the attribute of a document component which was in agreement with collation of said collation means based on the information about the designated attribute. It is characterized by the above-mentioned.

【0029】**[0029]**

【作用】

第1の発明では、解釈手段によって解釈された文書構造パターンと、再編成手段によって再編成された構造化文書とを照合手段によって照合し、更に出力処理手段が、その照合により一致した文書構成要素を再編成された構造化文書から抽出し出力するようにしたので、構造化文書から、指定された階層構造に従った文書構成要素を検索し出力することができる。

【0030】

第2の発明では、第1の発明において、出力処理手段は、照合手段の照合により一致した文書構成要素に対して、指定手段により指定された所定の処理例えば削除、置換、複写などの処理を施した後、出力するようにしているので、構造化文書から、指定された階層構造に従った文書構成要素に対して、削除、置換、複写などの処理を施すことができる。

【0031】

第3の発明では、第1の発明において、出力処理手段は、指定手段により削除処理が指定されると、蓄積手段に蓄積されてい

[OPERATION]

In 1st invention, the document structure pattern interpreted by interpretation means and the structuring document reorganized by reorganization means are compared by collation means, furthermore, since output-processing means extract from the structuring document which had the document component which was in agreement with the collation reorganized and it was made to output, the document component according to the designated hierarchical structure can be searched and outputted from a structuring document.

[0030]

In 2nd invention

After processing the fixed processing, for example, deletion, displacement, reproduction, etc., designated by designation means to the document component which was in agreement with collation of collation means, he is trying to output output-processing means in 1st invention.

Therefore, from a structuring document, deletion, displacement, reproduction, etc. can be processed to the document component according to the designated hierarchical structure.

[0031]

In 3rd invention

In 1st invention, output-processing means, if deletion processing is designated by designation means, while removing the

る、照合手段の照合により一致した文書構成要素の親である文書構成要素から、蓄積手段に蓄積されている照合手段の照合により一致した文書構成要素に関する情報（例えば文書構成要素を示すノード、そのノードの位置情報）を取り除くと共に、当該親である文書構成要素から削除されない、当該親の文書構成要素の子供である文書構成要素を出力するようにしているので、削除すべき文書構成要素の親の文書構成要素の内容を自動的に変更することができる。

information (for example, the node which shows a document component, the positional information on the node) about the document component which was in agreement with collation of the collation means accumulated in accumulation means from the document component which is the parents of the document component which was in agreement with collation of collation means accumulated in accumulation means, the document component which is not deleted from the document component which is said parents and which is the child of said parents' document component is outputted.

Therefore, the content of the document component of the parents of a document component who should delete can be altered automatically.

【0032】

第4の発明では、解釈手段が、基準となる文書構成要素同志の接続関係を解釈し、また再編成手段が、格納手段に格納されているファイル内の複数の構造化文書それぞれを照合処理可能な形式の構造に再編成し、また照合手段が、解釈手段による解釈結果と、再編成手段による再編成結果とを照合し、更に出力処理手段が、照合手段の照合により一致した文書構成要素を前記再編成結果から抽出するようにしているので、複数の構造化文書から、指定された階層構造に従った文書構成要素を検索し出

[0032]

In 4th invention, interpretation means interpret the relation of connection of the document component element who becomes a reference standard, moreover, reorganization means reorganize two or more structuring documents of each in the file stored in storing means in the structure of the form which can carry out collation processing, moreover, collation means compare the interpretation result by interpretation means, and the reorganization result by reorganization means, furthermore, output-processing means extract the document component which was in agreement with collation of collation means from said reorganization result.

Therefore, from two or more structuring

力することができる。

documents, the document component according to the designated hierarchical structure can be searched and outputted.

【0033】

第5の発明では、第4の発明において、複数の構造化文書を保存した複数のファイルを対象として、文書構成要素同志の接続関係に適合する文書構成要素を再編成結果から抽出するようにしているので、複数のファイルそれぞれに保存されている複数の構造化文書から、指定された階層構造に従った文書構成要素を検索し出力することができる。

[0033]

In 5th invention

In 4th invention, the document component which adapts a document component element's relation of connection is extracted from a reorganization result by making into object two or more files which conserved two or more structuring documents.

Therefore, from two or more structuring documents conserved at two or more files of each, the document component according to the designated hierarchical structure can be searched and outputted.

【0034】

第6の発明は、第4の発明において、ソース側及びターゲット側それぞれの複数の構造化文書を保存した複数のファイルを対象として、文書構成要素同志の接続関係に適合する文書構成要素を再編成結果から抽出すると共に、ターゲット側の抽出結果である単数又は複数の文書構成要素に対するソース側の抽出結果である単数又は複数の文書構成要素の挿入を実行するようにしているので、ターゲット側における複数のファイルそれぞれに保存されている複数の構造化文書から抽出された単数又は文書構成要素に対して、ソース側

[0034]

In 6th invention

In 4th invention, while extracting the document component which adapts a document component element's relation of connection by making into object two or more files as for which the source and target side conserved each structuring documents of two or more from a reorganization result, it is made to perform insertion of the document component of the unit or plurality which is an extraction result by the side of the source with respect to the document component of the unit or plurality which is an extraction result by the side of a target.

Therefore, the unit or document component extracted from two or more structuring documents conserved at two or more file それぞ which can be set to the source side can be

における複数のファイルそれぞれに保存されている複数の構造化文書から抽出された単数又は文書構成要素を一度に挿入することができる。

inserted at once to the unit or document component extracted from two or more structuring documents conserved at two or more files of each which can be set to a target-

【0035】

第7の発明では、第2の発明乃至第6の発明において、出力処理手段は、指定された属性に関する情報に基づいて、照合手段の照合により一致した文書構成要素の属性の参照又は変更の操作を実行するようにしているので、構造化文書から、指定された階層構造に従った文書構成要素を検索し、この文書構成要素の属性に対する参照又は変更の操作を実施することができる。

[0035]

In 7th invention, in 2nd invention or 6th invention, output-processing means are made to perform the operation of refer or alteration of the attribute of a document component which was in agreement with collation of collation means based on the information about the designated attribute.

Therefore, the document component which followed the designated hierarchical structure from the structuring document is searched, the operation of refer or alteration with respect to the attribute of this document component can be implemented.

【0036】**【実施例】**

以下、第1の実施例乃至第5の実施例を添付図面を参照して説明する。

[0036]**[EXAMPLES]**

Hereafter, 1st Example through 5th Example is demonstrated with reference to an accompanying drawing.

【0037】

最初に第1の実施例を図1乃至図9を参照して説明する。

[0037]

1st Example is initially demonstrated with reference to FIG. 1 - FIG. 9.

【0038】

図1は、本発明に係る文書処理装置の第1の実施例を示す機能ブロック図である。

[0038]

FIG. 1 is a functional-block figure which shows 1st Example of the word-processing apparatus based on this invention.

【0039】

同図において、メモリ10には、基準となる文書構成要素同志の接続関係(階層関係や順序関係)のパターン記述情報20(これについては後述する)が記憶されており、解釈部30は、メモリ10からパターン記述情報20を読み出して解釈し、この解釈結果である文書構造パターン40(これについては後述する)をメモリ10に記憶する。再編成部50は、文書ファイル60に保存されている構造化文書内を走査して、その構造化文書を照合処理可能な形式の構造に再編成し、この再編成結果である再編成構造化文書70(これについては後述する)をメモリ10に格納する。そして照合部80は、メモリ10に記憶されている文書構造パターン40と再編成構造化文書70と照合し、この照合結果を出力処理部90に出力する。出力処理部90では、照合部80の照合により一致した文書構成要素を再編成構造化文書70から抽出し出力する。

【0040】

図2は、図1に示した実施例の装置を実現するためのハードウェア構成を示したものであり、

[0039]

In this figure, the pattern description information 20 (about this, it mentions later) on the relation of connection (hierarchical relationship and order relation) of the document component element who becomes a reference standard is stored in memory 10, the interpretation section 30 reads and interprets the pattern description information 20 from memory 10, the document structure pattern 40 (about this, it mentions later) which is this interpretation result is stored in memory 10.

The reorganization section 50 scans the inside of the structuring document conserved at the text file 60, the structuring document is reorganized in the structure of the form which can carry out collation processing, the reorganization structuring document 70 (about this, it mentions later) which is this reorganization result is stored in memory 10.

And the collation section 80 compares with the document structure pattern 40 and the reorganization structuring document 70 which are stored in memory 10, this collation result is outputted to the output-processing section 90.

In the output-processing section 90, the document component which was in agreement with collation of the collation section 80 is extracted from the reorganization structuring document 70, and is outputted.

[0040]

FIG. 2 showed the hardware composition for implementing the apparatus of the Example shown in FIG. 1.

例えば、ワークステーションやコンピュータ等のブロック図を示している。図2において、装置は、構造化文書に対する検索処理を実行する中央処理装置（以下、CPUという）210と、主メモリ220と、ディスク230と、各種のデータ内容や文書内容を表示するディスプレイ240と、キーボードやマウスから構成され各種データや指令を入力する入力装置250とがバス260を介してそれぞれ接続されている。なおCPU210はバス260を介してこれに接続された各部を制御する。

For example, block diagrams, such as a workstation and a computer, are shown.

In FIG. 2, as for the apparatus, the central processing unit (henceforth CPU) 210 which performs search processing with respect to a structuring document, main memory 220, disc 230, display 240 that displays various kinds of content of data and content of a document, and the input device 250 which comprises a keyboard and a mouse and inputs various data and a command are connected through bus 260, respectively.

In addition, CPU210 controls each part connected to this through bus 260.

【0041】

ここで、図1に示した機能ブロック図の構成要素と図2に示したブロック図の構成要素との対応関係について説明する。図1に示したメモリ10は主メモリ20に対応しており、図1に示した解釈部30、再編成部50、照合部80、および出力処理部90は共にCPU210に対応しており、文書ファイル60はディスク230に対応している。

[0041]

Here, the correspondence of the component of the functional-block figure shown in FIG. 1 and the component of a block diagram shown in FIG. 2 is demonstrated.

Memory 10 shown in FIG. 1 is corresponding to main memory 20, both the interpretation section 30 shown in FIG. 1, the reorganization section 50, the collation section 80, and the output-processing section 90 are corresponding to CPU210, a text file 60 is corresponding to disc 230.

【0042】

次に上述したパターン記述情報20について説明する。

[0042]

Next, the pattern description information 20 mentioned above is demonstrated.

【0043】

パターン記述情報20は、パターンが、“節”、“表題”などの単純文字列パターン、“/”、“#”などの接続表現記号、“(”、“)”などの括弧、等の構文要素が特定の文法に従って出現するように表現されるものであり、例えば、

(節/表題/introduction) # (節/表題)
... (1)

ここで、“/”は包含関係、“#”は順序関係を表している。のように記述される。この記述内容はメモリ10に記憶される。

【0044】

そして解釈部30は、メモリ10から上記(1)に示す様なパターン記述情報を読み出して解釈し、この結果として図3に示す様に階層構造(木構造)を形成している文書構造パターン(これが上述した文書構造パターン40に相当する)を生成する。このとき、当然、上述したような構文要素や文法などが考慮されて、文書構造パターンが生成されることとなる。なおこの実施例では、図3に示すような文書構造パターンにおける矩形図形を単純パターンということにする。

[0043]

The pattern description information 20 is expressed so that a pattern may appear according to a grammar with specific syntax components, such as simple character string patterns, such as a "clause" and a "title", connection expression symbols, such as "/" and "#", and such as brackets, "(", ")".

For example, (Clause / title / introduction) # (Clause/title)... (1)

Here, "/" expresses an inclusive relation and "#" expresses order relation.

It describes like these.

This descriptive content is stored in memory 10.

[0044]

And the interpretation section 30 reads and interprets pattern description information as shown from memory 10 to said (1), the document structure pattern (it carries out considerable to the document structure pattern 40 which this mentioned above) which forms hierarchical structure (tree structure) as this result as shown in FIG. 3 is generated.

At this time, naturally it will consider the above-mentioned syntax component, the above-mentioned grammar, etc., and a document structure pattern will be generated.

In addition, in this Example, the rectangle figure in a document structure pattern as shown in FIG. 3 is made a simple pattern.

【0045】

ここで、解釈部30によるパターン記述の解釈処理について、図4に示すフローチャートを参照して説明する。なおここでは、括弧("("、")")の構文要素の処理を省略している。

[0045]

Here, interpretation processing of the pattern description by the interpretation section 30 is demonstrated with reference to the flowchart shown in FIG 4.

In addition, processing of a bracket ("(", ")") syntax component is omitted.

【0046】

解釈部30は、カレントレコードを生成し(ステップ401)、その後、入力文字列(例えば上記(1)のパターン記述情報)についての解釈は終りか否かを判断する(ステップ402)。入力文字列についての解釈処理がまだ残っている場合は、次の文字が順序関係を表す記号"#"か否かを判断する(ステップ403)。

[0046]

The interpretation section 30 generates a current record (step 401), after that, it is judged whether the interpretation about an input string (for example, pattern description information on said (1)) is end (step 402).

When a remaining still requires the interpretation processing about an input string, the next character judges whether it is a symbol "#" showing order relation (step 403).

【0047】

ステップ403において記号"#"であれば、新たなレコードを生成し、このレコードをカレントレコードの弟にし(ステップ404)、その後、新たに生成したレコードをカレントレコードと定義する(ステップ405)。その後、上記ステップ402に戻りこのステップ以降を実行する。

[0047]

A new record will be generated if it is a symbol "#" in step 403, this record is made into the younger brother of a current record (step 404), and the record newly generated after that is defined as a current record (step 405).

After that, it returns to the above-mentioned step 402, and this step or subsequent ones is performed.

【0048】

ステップ403において文字が記号"#"でない場合は、当該

[0048]

When a character is not a symbol "#" in step 403, it judges whether it is the symbol "/" with

文字が包含関係を表す記号“/”か否かを判断し（ステップ406）、記号“/”の場合は、新たなレコードを生成し、このレコードをカレントレコードの子にする（ステップ407）。その後、ステップ405に進む。

which said character expresses an inclusive relation (step 406), and, in the case of a symbol "/", a new record is generated, this record is made into the child of a current record (step 407).

After that, it progresses to step 405.

【0049】

ステップ406において文字が記号“/”でない場合は、文字であることを意味するので、当該文字をカレントレコード内に挿入し（ステップ408）、その後、ステップ402に戻りこのステップ以降を実行する。

[0049]

When a character is not a symbol "/" in step 406, it means that it is a character.

Therefore, said character is inserted into a current record (step 408), after that, it returns to step 402 and this step or subsequent ones is performed.

【0050】

なお、ステップ402において入力文字列についての解釈が終了した場合は処理を終了する。

[0050]

In addition, processing is completed when the interpretation about an input string is completed in step 402.

【0051】

ここで、具体例を挙げて説明する。例えば“富士夫/太郎#花子”という文字列は、図5(a)～(j)に示すようにパターンとして解釈されていく。なお、同図において、矩形図形がレコードを表している。また図5(j)に示す内容が最終的な文書構造パターンである。

[0051]

Here, an example is given and demonstrated.

For example, the character string "the Fujio / Taro # Hanako" is interpreted as a pattern as shown in FIG.5(a)-(j).

In addition, in this figure, the rectangle figure expresses the record.

Moreover, the content shown in FIG. 5 (j) is a final document structure pattern.

【0052】

上述した例では、文字列のパターン解釈であったが、図形エデ

[0052]

In the example mentioned above, it was the pattern interpretation of a character string.

ィタを用いて描画したグラフ（グラフ理論におけるグラフ）を解釈してパターンとする方法もある。このときは、ノードやリンクを適切な意味に解釈するように定義する。例えば、図 3 に示した例では、矩形で囲まれた文字列（例えば節や表題）が単純文字列パターンを表し、上下の矩形図形を結んでいるリンク（例えば符号 3 0 1 で示す線分（リンク））が序列関係を表し、左右の矩形図形を結んでいるリンク（例えば符号 3 0 2 で示す線分（リンク））が包含関係を表している。このような図形から意味構造を抽出するには、例えばパターン記述専用の図形エディタを用意すれば良い。

However, there is also the method of interpreting the diagrammatic chart (diagrammatic chart in a graph theory) which drew using the figure editor, and using as a pattern.

At this time, it defines as interpreting a node and a link to the suitable meaning.

For example, the character string (for example, a node and a title) surrounded with the rectangle expresses a simple character string pattern with the example shown in FIG. 3, the link (for example, line segment shown with code 301 (link)) to which the up-and-down rectangle figure is connected expresses a rank relationship, the link (for example, line segment shown with code 302 (link)) to which the rectangle figure on either side is connected expresses the inclusive relation.

What is sufficient is just to prepare the figure editor for example, only for pattern descriptions, in order to extract a semantic structure from such a figure.

【0053】

ここで、図形エディタを用いたパターン記述の一例を図 6 に示す。

[0053]

Here, an example of a pattern description using a figure editor is shown in FIG. 6.

【0054】

まずユーザは、図 6 (a) に示す様にパターンエディタの初期画面つまりウィンドウ 6 0 0 を表示画面に表示せさ、次に図 6 (b) に示す様にノード“unspecified node”をマウス（入力装置 2 5 0 に設けられて

[0054]

First, a user displays the initial-screen stuffing window 600 of a pattern editor on the display screen, as shown in FIG.6(a), next, as shown in FIG.6(b), a mouse (provided in the input device 250) is operated, and a node "unspecified node" is chosen, after that, fixed operation is performed, and as shown in FIG.6(c), a popup

いる)を操作して選択し、その後、所定の操作を行って、図6(c)に示す様にポップアップメニュー610を表示させる。そして、ポップアップメニュー610の“set string”の項目を選択して、図6(d)に示す様に文字列を記述する。続いて、ポップアップメニュー610の“make child”の項目を選択して、図6(e)に示す様にノード“節”の子ノードを作成する。引き続いて、ポップアップメニュー610の“make brother”の項目を選択して、図6(f)に示す様に弟ノード作成する。こうして作成された図6(f)に示す様なグラフは直接文書構造パターンとして用いられる。すなわち、図6(f)に示す内容が、パターン記述情報20であり、また文書構造パターン40でもある。

【0.0.5.5】

次に、再編成部50により再編成される再編成構造化文書70について説明する。

【0.0.5.6】

ファイルとして保管されている文書のままでは、文書内部の構造へのアクセスができずパターン処理に不利なので、ファイルを走査して構造を再編成する。ただし、一度にファイル全体を

menu 610 is displayed.

And the item of "set string" of a popup menu 610 is chosen, a character string is described as shown in FIG.6(d).

Then, the item of "make child" of a popup menu 610 is chosen, as shown in FIG.6(e), the child node of a node "clause" is made.

Then, the item of "make brother" of a popup menu 610 is chosen, as shown in FIG.6(f), younger brother node creation is carried out.

In this way, a diagrammatic chart as shown in made FIG.6(f) is used as a direct document structure pattern.

That is, the content shown in FIG.6(f) is the pattern description information 20.

Moreover, it is also the document structure pattern 40.

[0055]

Next, the reorganization structuring document 70 reorganized by the reorganization section 50 is demonstrated.

[0056]

With the document currently stored as a file, the access to the structure inside a document cannot be performed, but since it is disadvantageous for pattern processing, a file is scanned and the structure is reorganized.

However, it is not necessary to analyze the

解析する必要はなく、照合部80が必要するとき、必要となっている部分のみを解析して出力するようにする。

【0057】

この解析処理としては、ファイルの必要とする箇所にファイルポインタを移動して可変長レコードを切り出し、そのレコードに予め記述されている構造上で位置情報をもとに木構造（或いは部分木構造）を再構成するようになっている。再編成して得られた再編成構造化文書の一例を図7に示す。この図に示す様に文書は、階層構造（木構造）として表現される。なおこの実施例では、再編成構造化文書における矩形図形を文書ノードということにする。

【0058】

次に、照合部80による照合処理について、図8に示すフローチャートを参照して説明する。

【0059】

照合部80は、current-nodeを、再編成構造化文書の構造における最初の文書ノードにし（ステップ801）、current-patを、文書構造パターンの構造における最初の単純パターンにする（ステップ802）。

whole file at once, only a needed part is analyzed and outputted when the collation section 80 carries out the need.

[0057]

As this analysis processing, a file pointer is transferred to the location which a file needs, and a variable-length record is segmented, a tree structure (or partial tree structure) is reconfigured based on the positional information on the structure beforehand described by the record.

An example of the reorganization structuring document reorganized and obtained is shown in FIG. 7.

As shown to this figure, a document is expressed as hierarchical structure (tree structure).

In addition, in this Example, the rectangle figure in a reorganization structuring document is made a document node.

[0058]

Next, the collation processing by the collation section 80 is demonstrated with reference to the flowchart shown in FIG. 8.

[0059]

The collation section 80 makes current-node the first document node in the structure of a reorganization structuring document (step 801), and uses current-pat as the first simple pattern in the structure of a document structure pattern (step 802).

【0060】

その文書ノードおよび単純パターンは指定された接続条件に一致するか否かを判断し（ステップ803）、一致する場合は、current-pat と current-node とが一致するか否かを判断する（ステップ804）。

[0060]

It judges whether the document node and simple pattern are in agreement with the designated connection conditions (step 803), and when in agreement, it is judged whether current-pat and current-node are in agreement (step 804).

【0061】

ステップ804において一致する場合は、current-node を次の文書ノードにし（ステップ805）、その後、単純パターンが終りか否かを判断する（ステップ806）。

[0061]

When in agreement in step 804, current-node is made into the next document node (step 805), and, after that, it is judged whether a simple pattern is end (step 806).

【0062】

ステップ806において単純パターンが終了した場合は、その旨が照合部80から出力処理部90に通知される。出力処理部90では、その一致した文書ノードを出力する（ステップ807）。このようにして出力される文書ノードは、ディスプレイ240に表示されたり、あるいはファイルとして保管される。さらには、その文書ノードを他の（或いは同一の）文書中の特定領域に流し込むことによって、文書内容を編集することができる。このように検索して一致した文書ノードすなわち文書構成要素を、削除や置換したり、他

[0062]

When a simple pattern is completed in step 806, the output-processing section 90 notifies that from the collation section 80.

The congruous document nodes are outputted in the output-processing section 90 (step 807). Thus, the document node outputted is displayed on display 240.

Or it is stored as a file.

Furthermore, the content of a document can be edited by slushing the document node into the specific region in other documents (or it being the same).

Thus, it searches, and it can delete and replace, congruous document nodes, i.e., document component, and it can be inserted in other documents.

の文書に挿入することができる。

【0063】

ところでステップ807が終了すると、照合部80は、current-pat を前の単純パターンにし(ステップ808)、その後、文書ノードが終りか否かを判断す(ステップ809)。

[0063]

By the way, after step 807 is completed, the collation section 80 uses current-pat as a previous simple pattern (step 808), and, after that, judges whether a document node is end (step 809).

【0064】

文書ノードが終了したら処理を終了し、まだ文書ノードがある場合は上記ステップ803に戻りこのステップ以降を実行する。

[0064]

Processing will be completed if a document node is completed, when there is still a document node, it returns to the above-mentioned step 803, and this step or subsequent ones is performed.

【0065】

なお、ステップ806において単純パターンが終りの場合はステップ809に進む。

[0065]

In addition, when a simple pattern is end in step 806, it progresses to step 809.

【0066】

また上記ステップ803において指定された接続条件に不一致の場合、ステップ804において一致しない場合は、current-pat は最初の単純パターンか否かを判断し(ステップ810)、単純パターンであれば、一致していた文書ノードまで戻り(ステップ811)、その後、ステップ805に進む。

[0066]

Moreover, when disagreeable on the connection conditions designated in the above-mentioned step 803, when not in agreement in step 804, current-pat judges whether it is the first simple pattern (step 810), if it is a simple pattern, will return to the congruous document nodes (step 811), and, after that, will progress to step 805.

【0067】**[0067]**

ステップ 810において単純パターンでない場合は current-pat を前の単純パターンにし (ステップ 812)、その後、ステップ 811に進む。

When it is not a simple pattern in step 810, current-pat is used as a previous simple pattern (step 812), and after that, it progresses to step 811.

【0068】

以上のような処理を行って得られた照合結果を図 9 に示す。この図に示した例では、図 3 に示した文書構造パターンと図 7 に示した再編成構造化文書とを照合した場合の結果を示している。

[0068]

The collation result obtained by performing the above processing is shown in FIG. 9.

In the example shown to this figure, the result at the time of comparing the document structure pattern shown in FIG. 3 and the reorganization structuring document shown in FIG. 7 is shown.

【0069】

この例においては、図 9 中点線で示されるように、単純パターン 901A と文書ノード 901B とが一致し、単純パターン 902A と文書ノード 902B とが一致し、単純パターン 903A と文書ノード 903B とが一致し、単純パターン 904A と文書ノード 904B とが一致し、単純パターン 905A と文書ノード 905B とが一致し、単純パターン 906A と文書ノード 906B とが一致している。

[0069]

In this example, as it shows by the FIG. 9 middle point line, simple pattern 901A and document node 901B are in agreement, simple pattern 902A and document node 902B are in agreement, simple pattern 903A and document node 903B are in agreement, simple pattern 904A and document node 904B are in agreement, simple pattern 905A and document node 905B are in agreement, simple pattern 906A and document node 906B are in agreement.

【0070】

この図から分かるように、文書構造パターン (階層構造情報) が分かれば、例えば、単純パターン 906A の文字列が分から

[0070]

If a document structure pattern (hierarchical-structure information) is known as this figure shows, even if it is the case where the character string of simple pattern 906A is

ない場合であっても、照合処理することにより、その単純パターン906Aに対応する文書ノード906Bを得ることができ、その結果として、表題は「魔神の宅配便」であるということが分かる。

【0071】

また単純パターン903Aの文字列「i n t r o d u c t i o n」に一致するところが、文書ノード903B、907に存在していた場合であっても、必ず表題になっているもののみしか一致しないので（この例では文書ノード903Bのみ一致）、確実に検索することができる。

【0072】

以上説明したように第1の実施例によれば、文字列パターンに加えて、適切な方法で文書の構成要素の接続関係を示すことによって、誤りなく必要とするものが得られる。

【0073】

次に第2の実施例を図10乃至図15を参照して説明する。

【0074】

図10は、本発明に係る文書処理装置の第2の実施例を示す機能ブロック図である。この機能ブロック図は、図1に示した第

not known, document node 906B corresponding to that simple pattern 906A can be obtained by carrying out collation processing, for example, as a result, it turns out that a title is "the parcel delivery service of fiend."

[0071]

Moreover, since only the thing which is surely a title is in agreement even if the place which is in agreement with the character string "introduction" of simple pattern 903A is the case where it exists in the document nodes 903B and 907 (only document node 903B was in agreement in this example), it can search reliably.

[0072]

As explained above, according to 1st Example, in addition to a character string pattern, what is needed is obtained by showing the relation of connection of the component of a document by a suitable method without mistake.

[0073]

Next, 2nd Example is demonstrated with reference to FIG. 10 - FIG. 15.

[0074]

FIG. 10 is a functional-block figure which shows 2nd Example of the word-processing apparatus based on this invention. This functional-block figure has the composition

1の実施例の機能ブロック図の構成において、適合ノード蓄積部1010、命令処理部1020を追加し、出力処理部90を出力処理部1030に変更した構成になっている。なお図10において、図1に示した構成要素と同様の機能を果たす部分には同一の符号を付している。

【0075】

適合ノード蓄積部1010は、照合部80の照合結果である文書構成要素を蓄積する。

【0076】

命令処理部1020は、コマンドラインあるいは標準入力から与えられる所定の処理を解釈し、この解釈結果を出力処理部1030に与える。なお所定の処理には、“挿入する”、“置換する”、“削除する”の処理が含まれている。

【0077】

出力処理部1030は、適合ノード蓄積部1010に蓄積されている文書構成要素に対して、命令処理部1020からの処理命令に従って処理を実行し、出力する。この出力は、次の処理のための標準出力に出力しても良い。

of having added the compatible node storage part 1010 and the command-processing processing section 1020, and having altered the output-processing section 90 into the output-processing section 1030, in the composition of the functional-block figure of 1st Example shown in FIG. 1.

In addition, in FIG. 10, the same code is attached into the part which achieves the function similar to the component shown in FIG. 1.

[0075]

The compatible node storage part 1010 accumulates the document component which is the collation result of the collation section 80.

[0076]

The command-processing section 1020 interprets the fixed processing which it imparts from a command line or standard input, and gives this interpretation result to the output-processing section 1030.

In addition, processing of "inserting", "replacing", and "deleting" is included in fixed processing.

[0077]

The output-processing section 1030 performs processing to the document component accumulated in the compatible node storage part 1010 according to the processing command from the command-processing section 1020, and outputs.

It may output this output to the standard output for the next processing.

【0078】

なお上記標準入力及び標準出力とは、UNIX（オペレーティングシステム）における標準入力及び標準出力のことである。

[0078]

In addition, above-mentioned standard input and a standard output are standard input and the standard output in UNIX (operating system).

【0079】

この図10に示した装置も、図2に示した第1の実施例のハードウェア構成で実現することができる。ここで図10に示した構成要素と図2に示した構成要素との対応関係について説明する。図10に示した適合ノード蓄積部1010は図2に示した主メモリ220に対応し、図10に示した命令処理部1020及び出力処理部1030は共に図2に示したCPU210に対応している。他の構成要素については第1の実施例と同様である。

[0079]

The apparatus shown in this FIG. 10 is also realizable with the hardware composition of 1st Example shown in FIG. 2.

The correspondence of the component shown in FIG. 10 here and the component shown in FIG. 2 is demonstrated.

The compatible node storage part 1010 shown in FIG. 10 corresponds to the main memory 220 shown in FIG. 2, both the command-processing sections 1020 and output-processing sections 1030 that were shown in FIG. 10 are corresponding to CPU210 shown in FIG. 2.

About other components, it is the same as that of 1st Example.

【0080】

この第2の実施例は、基本的には第1の実施例と同様である。第1の実施例と異なるのは、構造化文書中から、文書構造パターンに一致する構造を抽出し、この抽出した構造に対して、“挿入”、“置換”、“削除”などの処理を施すという点である。

[0080]

This 2nd Example is the same as 1st Example fundamentally.

Differing from 1st Example extracts the structure which is in agreement with a document structure pattern out of a structuring document, it is the point of view of processing "insertion", "displacement", "deletion", etc. to this extracted structure.

【0081】

そこで、第2の実施例における

[0081]

Then, the text editing processing in 2nd

文書編集処理について、図 1.1 乃至図 1.5 を参照して説明する。

Example is demonstrated with reference to FIG. 11 - FIG. 15.

【0082】

パターン記述情報 20 として、
 (節／本文段落／スタミナ
 X) # 注
 ... (2)

ここで、／は包含関係を示す記号

は順序関係を示す記号

は処理対象となるノードを示す記号

が記述されメモリ 10 に記憶されている。

[0082]

As the pattern description information 20, (Clause / this stage / stamina X) #^notes... (2)

Here, / is a symbol which shows an inclusive relation.

The symbol which shows order relation

^ is the symbol which shows the node used as a process target.

These are described and it stores in memory 10.

【0083】

次に解釈部 30 は、図 4 に示した第 1 の実施例のパターン解釈処理手順と同様の処理を実行して、上記 (2) に示すパターン記述情報 20 から、図 1.1 に示す様な文書構造パターンを生成し、これを文書構造パターン 40 としてメモリ 10 に格納する。このとき処理対象となるノードは「注」であると解釈する。図 1.1 では、処理対象となるノードの目印として二重枠で囲んで表記している。この場合も、図 5 に示した第 1 の実施例の具体例の様にパターン解釈されていく。

[0083]

Next, the interpretation section 30 performs processing similar to the pattern interpretation procedure of 1st Example shown in FIG. 4, from the pattern description information 20 shown to said (2), a document structure pattern as shown in FIG. 11 is generated, it stores in memory 10 by making this into the document structure pattern 40.

The node which constitutes a process target at this time is interpreted as their being "notes."

In FIG. 11, it surrounds and writes by the double plate as a mark of the node used as a process target.

The pattern interpretation is carried out like the example of 1st Example shown in FIG. 5 also in this case.

【0084】

[0084]

この第2の実施例でも、図形エディタを用いて描画したグラフ（グラフ理論におけるグラフ）を解釈して文書構造パターンを認識することができる。図形エディタを用いたパターン記述の方法は、図6を用いて説明した第1の実施例と同様である。

Also in this 2nd Example, the diagrammatic chart (diagrammatic chart in a graph theory) which drew using the figure editor can be interpreted, and a document structure pattern can be recognized.

The method of a pattern description using a figure editor is the same as that of 1st Example demonstrated using FIG. 6.

【0085】

一方、再編成部50による再編成処理結果は、図12に示す内容であり、メモリ10に再編成構造化文書70として格納される。

[0085]

On the other hand, the reorganization processing result by the reorganization section 50 is content shown in FIG. 12.

It is stored in memory 10 as a reorganization structuring document 70.

【0086】

そして照合部80が、図8に示した第1の実施例の照合処理手順と同様の処理を実行して、図11に示す文書構造パターンと、図12に示す再編成構造化文書とを照合する。この結果として、図13に示すような照合結果が得られることとなる。図13においては、単純パターン1310と文書ノード1310A、1310Bとが一致し、単純パターン1320と文書ノード1320A、1320Bとが一致し、単純パターン1330と文書ノード1330A、1330Bとが一致し、単純パターン1340と文書ノード1340A、1340Bとが一致している。このとき照合部80は、パ

[0086]

And the collation section 80 performs processing similar to the collation procedure of 1st Example shown in FIG. 8, the document structure pattern shown in FIG. 11 and the reorganization structuring document shown in FIG. 12 are compared.

As this result, a collation result as shown in FIG. 13 will be obtained.

In FIG. 13, the simple pattern 1310 and the document nodes 1310A and 1310B are in agreement, the simple pattern 1320 and the document nodes 1320A and 1320B are in agreement, the simple pattern 1330 and the document nodes 1330A and 1330B are in agreement, the simple pattern 1340 and the document nodes 1340A and 1340B are in agreement.

At this time, the collation section 80 stores in the compatible node storage part 1010 the node

ターン記述情報中で指示されている処理対象ノードに対応する構造化文書中のノードを適合ノード蓄積部 1 0 1 0 に格納する。この例では「注」のノード 1 3 4 0 A、1 3 4 0 B が蓄積される。なおノードの全情報を格納する必要はなく、ノードへのリンクを記憶するようにしても良い。

in the structuring document corresponding to the process-target node currently indicated in pattern description information.

In this example, nodes 1340A and 1340B of "notes" are accumulated.

In addition, it is not necessary to store all the information on a node, and it may make it store the link to a node.

【0087】

命令処理部 1 0 2 0 に、例えば「insert “スタミナ X は絶倫製薬の登録商標です” as lastChild」というコマンドラインが与えられると、命令処理部 1 0 2 0 は、そのコマンドラインを、出力処理部 1 0 3 0 の動作を決定する変数として解釈し、この解釈結果に応じた処理を実行する。この処理を図 1 4 を用いて説明する。

[0087]

If it imparts the command line "insert "Stamina X is the trademark of a Zetsurin pharmaceutical" as last Child " to the command-processing section 1020, the command-processing section 1020 interprets the command line as a variable which determines an operation of the output-processing section 1030, processing according to this interpretation result is performed.

This processing is demonstrated using FIG. 14.

【0088】

最初に insert を解釈して、“挿入する”を示す insert のコマンドを出力処理部 1 0 3 0 に設定する。コマンドの設定は、解釈したコマンドを設定するようにしても良いし、出力処理部 1 0 3 0 に予め設定されたコマンド群を用意しておき、解釈したコマンドに該当するものを設定するようにしても良い。ここでは、後者の方法を採用しており、“挿

[0088]

Insert is interpreted initially, the command of insert which shows "inserting" is set as the output-processing section 1030.

It may make it a setup of a command set up the interpreted command.

The predetermined command group is prepared for the output-processing section 1030, and it may make it set up the thing applicable to the interpreted command.

Here, the latter method is adopted, the commands insert, replace, and delete which

入する”、“置換する”、“削除する”にそれぞれ対応するコマンド insert, replace, delete が予め用意されている。これらのコマンド以外にも、insert from file"tottekoi", insert from stdin, remove などのコマンドを用意することもできる。

respectively correspond to "inserting", "replacing", and " deleting" are prepared beforehand.

Besides these commands, insert from file"tottekoi", insert from stdin, and a remove etc. command can also be prepared.

【0089】

次に“スタミナXは絶倫製薬の登録商標です”を解釈して、処理上必要となるデータ領域（ここでは buffer という名前で参照される）に文字列を出力処理部 1030 に複写する。

[0089]

Next, it interprets "Stamina X is the trademark of Zetsurin pharmaceutical", a character string is reproduced to the data area (here, seen by the name of buffer) which is needed on processing at the output-processing section 1030.

【0090】

最後に as lastChild を解釈して、末子であるということを示すフラグ 1 を出力処理部 1030 に設定する。

[0090]

Finally as last Child is interpreted, flag 1 which shows that she is last-born child is set as the output-processing section 1030.

【0091】

出力処理部 1030 では、「insert のコマンド」、「buffer =スタミナXは絶倫製薬の登録商標です」、「フラグ 1」が設定されると、これらの情報に基づいて、適合ノード蓄積部 1010 に蓄積されているノードに対して処理を施し、この結果をファイルストリームに出力する。この出力結果を図 15 に示す。この図 15 に示す例では、注のノード 1340A、1340B

[0091]

In the output-processing section 1030

If "the command of insert", "the buffer= stamina X is the trademark of Zetsurin pharmaceutical", and "flag 1" are set up, it processes to the node accumulated in the compatible node storage part 1010 based on such information, this result is outputted to a file stream.

This output result is shown in FIG. 15.

In the example shown in this FIG. 15, nodes 1500A and 1500B of "Stamina X is the trademark of Zetsurin pharmaceutical" are connected as a child of nodes 1340A and

の子供として“スタミナXは絶倫製薬の登録商標です”のノード1500A、1500Bが接続されている。なお処理と出力は同時に実行することができる。更には対象ノードの照合の検証とも同時に実行することができる。

【0092】

以上説明したように第2の実施例によれば、構造化文書内のオブジェクト間の関係を利用したパターン、つまり基準となる階層構造のパターンとのパターンマッチングを行うようにしているので、構造化文書における正確な情報（文書構成要素）にアクセスすることができると共に、階層上のデータ（文書構成要素）の位置を簡単に指定することができる。

【0093】

また1つのパターンを使用して上述したようなパターンマッチングを行うようにしているので、構造化文書中の複数の書き換え又は挿入位置を指定することができる。

【0094】

更にユーザとのインタラクションなしで正確なアクセスを実行することができるので、構造を持った文書のバッチ処理が可能

1340B of notes.

In addition, processing and an output can be performed simultaneously.

Furthermore, verification of collation of an object node can be performed simultaneously.

[0092]

As explained above, according to 2nd Example, it is made to perform a pattern matching with the pattern using the relationship between the objects in a structuring document, and the pattern of the hierarchical structure which constitutes a reference standard in other words. Therefore, while being able to access to the exact information (document component) in a structuring document, the position of the data on a hierarchy (document component) can be designated easily.

[0093]

Moreover, it is made to perform the above-mentioned pattern matching using one pattern.

Therefore, two or more overwrite or insertion points in a structuring document can be designated.

[0094]

Furthermore, an exact access can be performed without interaction with a user.

Therefore, the batch processing of a document with the structure can be performed.

となる。

【0095】

次に第3の実施例について、図16乃至図20を参照して説明する。

[0095]

Next, 3rd Example is demonstrated with reference to FIG. 16 - FIG. 20.

【0096】

図16は、本発明に係る文書処理装置の第3の実施例を示す機能ブロック図である。この機能ブロック図は、図10に示した第2の実施例の機能ブロック図の構成において、適合ノード蓄積部1010を削除し、削除情報蓄積部1040を追加した構成になっている。

[0096]

FIG. 16 is a functional-block figure which shows 3rd Example of the word-processing apparatus based on this invention.

This functional-block figure has the composition of having deleted the compatible node storage part 1010 and having added the deletion information storage part 1040, in the composition of the functional-block figure of 2nd Example shown in FIG. 10.

【0097】

削除情報蓄積部1040は、照合部80の照合により一致した文書ノードと、当該文書ノードの親である文書ノードとを対応付けして蓄積（この蓄積の構造については後述する）し管理する。

[0097]

The deletion information storage part 1040 matches the document node which was in agreement with collation of the collation section 80, and the document node which is the parents of said document node, and accumulates and manages it (about the structure of this accumulation, it mentions later).

【0098】

出力処理部1030は、命令処理部1020から“削除する処理”が渡されると、削除情報蓄積部1040に蓄積されている照合部80の照合により一致した文書ノードの親である文書ノードから、削除情報蓄積部1040に蓄積されている照合部80

[0098]

If "processing to delete" is passed to the output-processing section 1030 from the command-processing section 1020, while removing the information (for example, a document node, the positional information on the node) about the document node which was in agreement with collation of the collation section 80 accumulated in the deletion

0の照合により一致した文書ノードに関する情報（例えば文書ノード、そのノードの位置情報）を取り除くと共に、当該親の文書ノードから削除されない子供の文書ノードを出力する。

information storage part 1040 from the document node which is the parents of the document node which was in agreement with collation of the collation section 80 accumulated in the deletion information storage part 1040, a child's document node which is not deleted from said parents' document node is outputted.

【0099】

なお構造化文書においては、一般的に、各ノード間の関係を表現する際に、親ノードには自己の子供である子ノードを示す情報が含まれており、一方、子ノードに自己の親である親ノードを示す情報が含まれているので、出力処理部1030は、“削除する処理”を受け取ったときは、親の文書ノードから、削除すべき文書ノードを示す情報を削除するようにしている。このとき、削除される文書ノードに含まれている親の文書ノードを示す情報は削除してもしなくとも良い。但し、親の文書ノードから削除すべき文書ノードを示す情報を削除することにより、当該削除すべき文書ノードは、親の文書ノードとの関連性がなくなり出力されないので、処理効率の点からいって、削除される文書ノードに含まれている親の文書ノードを示す情報は削除しない方が良い。

[0099]

In addition, in the structuring document, when expressing the relationship between each node generally, the information which shows the child node which is a self child is included in the parent node, on the other hand, the information which shows the parent node which is self parents is included in the child node.

Therefore, when "processing to delete" is received, he is trying for the output-processing section 1030 to delete the information which shows the document node which should be deleted from parents' document node.

It may not be necessary to delete the information which shows parents' document node contained in the document node deleted at this time.

However, by deleting the information which shows the document node which should be deleted from parents' document node, the correlation with parents' document node of said document node which should be deleted is lost, and it is not outputted.

Therefore, it is better to say from a point of view of processing effectiveness, and not to delete the information which shows parents' document node contained in the document node deleted.

【0100】

図17は、削除情報蓄積部1040に蓄積される削除情報の構造の一例を示している。この実施例では、削除情報の構造を、削除される文書ノード（以下、子ノードという）のリスト（以下、子リストという）を持った、その親の文書ノード（以下、親ノードという）のリスト（以下、親リストという）として表現するようにしている。因みに図17においては、親リストには、2つの子ノードC1-1、C1-2の子リストを持つ親ノードP1と、2つの子ノードC2-1、C2-2の子リストを持つ親ノードP2とが登録されている。

[0100]

FIG. 17 is showing an example of the structure of the deletion information accumulated in the deletion information storage part 1040.

In this Example, the structure of deletion information is expressed as a list (henceforth a parent list) with the list (henceforth a child list) of document nodes (henceforth a child node) deleted of those parents' document nodes (henceforth a parent node).

Incidentally in FIG. 17, the parent node P1 with two child node C1-1, C1-2 child lists and the parent node P2 with two child node C2-1, C2-2 child lists are registered into the parent list.

【0101】

なお、上記の例では削除される子ノードを登録するようにしているが、削除される子ノードの情報としては、何番目の子供が削除されるべきか、という情報で十分である。

[0101]

In addition, the child node deleted is registered in the above-mentioned example.

However, as information on the child node deleted, the information the child of what position should be deleted is enough.

【0102】

次に、削除情報蓄積部1040による削除情報の作成処理について、図18に示すフローチャートを参照して説明する。

[0102]

Next, creation processing of the deletion information by the deletion information storage part 1040 is demonstrated with reference to the flowchart shown in FIG. 18.

【0103】

削除情報蓄積部1040は、初期化として親リストを空にし

[0103]

The deletion information storage part 1040 empties a parent list as initialization (step 1051),

(ステップ1051)、次に、照合部80から渡される照合結果つまりパターンと一致するノード(以下、これをノードCとする)を順次受け取ると共に、当該ノードCを1つ取り込む(ステップ1052)。

[0104]

次に削除情報蓄積部1040は、取り込んだノードCの親ノードPは親リストに未登録か否かを判断する(ステップ1053)。

[0105]

ステップ1053において親ノードPは登録済みの場合は、ノードCは親ノードPの子リストに未登録か否かを判断する(ステップ1054)。

[0106]

ステップ1054においてノードCは登録済みの場合は、照合部80からの次のノードCを取り込む(ステップ1055)。

[0107]

なお、ステップ1054においてノードCが未登録の場合は、ノードCを親ノードPの子リストに新規登録し(ステップ1056)、その後、ステップ1055に進み、またステップ1053において親ノードPが未登録

next, it is while receiving in order the node (let this be Node C hereafter) which is in agreement with the collation sequence stuffing pattern passed from the collation section 80, said one node C is received (step 1052).

[0104]

Next, the deletion information storage part 1040, it judges that parent node P of the received node C is about whether it is un-registering to a parent list (step 1053).

[0105]

In step 1053, as for parent node P, when registered, Node C judges whether it is un-registering on the child list of parent node P (step 1054).

[0106]

In step 1054, Node C receives the next node C from the collation section 80, when registered (step 1055).

[0107]

In addition, when Node C has not been registered in step 1054, Node C is newly registered into the child list of parent node P (step 1056), when it progresses to step 1055 and parent node P has not been registered in step 1053 after that, it newly registers with a parent list (step 1057), and, after that,

の場合は、親リストに新規登録し(ステップ1057)、その後、ステップ1054に進む。

【0108】

ところで上記ステップ1055を終了した場合、削除情報蓄積部1040は、照合部80から渡されるノードは終りか否かを判断し(ステップ1058)、ノードが終りの場合には処理を終了し、一方、まだノードが存在している場合は、上記ステップ1053に戻り、照合部80から渡されるノードが終了するまで、ステップ1053～1058を繰り返す。

【0109】

ここで、図17に示した例を用いて、削除情報の作成処理を説明する。

【0110】

図17において、親ノードP1が親ノードP2より早く出現するものとし、今現在、親リストは空き状態とする。

【0111】

このような状態で、上記ステップ1053において、子ノードC1-1が取り込まれた後、ステップ1053が実行された際には、子ノードC1-1の親ノードP1は未登録であるので、この

progresses to step 1054.

[0108]

By the way, when the above-mentioned step 1055 is completed, as for the deletion information storage part 1040, the node passed from the collation section 80 judges whether it is end (step 1058), processing is completed when a node is finally, on the other hand, when the node still exists, it returns to the above-mentioned step 1053, and step 1053-1058 is repeated until the node passed from the collation section 80 is completed.

[0109]

Here, creation processing of deletion information is demonstrated using the example shown in FIG. 17.

[0110]

In FIG. 17, a parent node P1 shall appear earlier than a parent node P2.

A parent list takes for a vacant state now.

[0111]

After child node C1-1 is received in the above-mentioned step 1053 in such the state, when step 1053 should perform, the parent node P1 of child node C1-1 has not registered. Therefore, it progresses to step 1057 in this case, and a parent node P1 is registered into a

場合はステップ1057に進み、このステップにより親ノードP1が親リストに登録される。このステップ1057終了後はステップ1054に移行するが、子ノードC1-1は親ノードP1の子リストには登録されていないので、ステップ1056に進み、このステップにより子ノードC1-1が親ノードP1の子リストに登録される。

[0112]

そしてステップ1055、1058が実行されることとなり、この場合は、照合部80からのノード（つまり照合結果であるノード）がまだ存在するので、上記ステップ1053に戻り、このステップにより子ノードC1-2についての処理が実行される。この場合は、ステップ1053においては「NO」（つまり親ノードP1は親リストに登録済み）となるので、ステップ1054に進み、このステップにおいては「YES」（子ノードC1-2は親ノードP1の親リストには未登録）なので、ステップ1056に進み、このステップにより子ノードC1-2が親ノードP1の親リストに登録される。

[0113]

以下同様に、子ノードC2-1、

parent list by this step.

After this step 1057 completion moves to step 1054.

However, since child node C1-1 is not registered into the child list of parent nodes P1, it progresses to step 1056 and child node C1-1 is registered into the child list of parent nodes P1 by this step.

[0112]

And steps 1055 and 1058 will be performed, in this case, the node (node which is a collation result in other words) from the collation section 80 still exists.

Therefore, it returns to the above-mentioned step 1053, and processing about child node C1-2 is performed by this step.

In this case, it is set to "NO" (in other words, a parent node P1 is registered to a parent list) in step 1053.

Therefore, it progresses to step 1054, and since it is "YES" (child node C1-2 is un-registering in the parent list of a parent node P1) in this step, it progresses to step 1056 and child node C1-2 is registered into the parent list of a parent node P1 by this step.

[0113]

Processing about child node C2-1 and C2-2 is

C 2-2 についての処理が行われる。

performed like the following.

【0114】

なお図 17 において、削除すべきノードが、子ノード C 1-1、C 1-2、親ノード P 1 であった場合は、親リストに、例えば親ノード P 1 が削除される旨の情報のみを登録し、子ノード C 1-1、C 1-2 については登録しないようにする。何故ならば、削除されるノードから更に削除されるということは無いので、親ノード P 1 が削除される旨のみを登録すれば良いこととなる。従って、子ノード C 1-1、C 1-2 のリストへの登録を行う必要がないので、処理効率及び記憶使用効率を向上させることができる。

[0114]

In addition, in FIG. 17, when the nodes which should be deleted are child node C1-1, C1-2, and a parent node P1, only the information on a purport that a parent node P1 is deleted is registered into a parent list, it does not register about child node C1-1 and C1-2.

Because, since it does not say that it is further deleted from the node deleted, what is necessary will be to register only the purport from which a parent node P1 is deleted.

Therefore, since it is not necessary to perform registration to child node C1-1 and a C1-2 list, processing effectiveness and memory use effectiveness can be improved.

【0115】

次に、出力処理部 103 の出力処理について、図 19 に示すフローチャートを参照して説明する。

[0115]

Next, output processing of the output-processing section 103 is demonstrated with reference to the flowchart shown in FIG. 19.

【0116】

出力処理部 103 は、命令処理部 1020 からの“削除する処理”を受け取ると（ステップ 1061）、処理の対象をファイル先頭に移動し（ステップ 1062）、その後、そのファイルから、文書のルートノード（これはフ

[0116]

If the output-processing section 103 receives "processing to delete" from the command-processing section 1020 (step 1061), the object of processing is transferred to a file head (step 1062), after that, it is reading (step 1063) about one node N which is a route node (this is a file head) of the file to a document,

ファイル先頭) であるノードNを1つ読み取り (ステップ1063)、該ノードNについて、出力すべきノードを出力するノード出力処理を実行する (ステップ1064)。すなわちステップ1064においては、ノードNの下位に存在するノードが出力されることになる。

node output processing which outputs the node which should be outputted about this node N is performed (step 1064).

That is, in step 1064, the node which exists in the lower-order of Node N is outputted.

【0117】

このステップ1064のノード出力処理について、図20に示すサブルーチンを参照して説明する。

[0117]

Node output processing of this step 1064 is demonstrated with reference to the subroutine shown in FIG. 20.

【0118】出力処理部103は、ノードNは親リストに登録されているか否かを判断する (ステップ1071)。ここで、ノードNはルートノードであるので、このノードNが、親リストに登録されているということは、必ず削除されるノード (すなわち子ノード) が存在することを意味しており、一方、親リストに登録されていない場合は、削除されるノードが存在しないことを意味している。

[0118]

The output-processing section 103 judges whether Node N is registered into the parent list (step 1071).

Here, Node N is a route node.

Therefore, that this node N is registered into the parent list is meaning that the node (that is, child node) surely deleted exists, on the other hand, when not registering with a parent list, it is meaning that the node deleted does not exist.

【0119】

ところで、ステップ1071において登録済みの場合は、ノードNについて変更を施して出力する (ステップ1072)。このステップ1072においては、削除情報蓄積部1040に蓄積

[0119]

By the way, in step 1071, when registered, it alters and outputs about Node N (step 1072).

In this step 1072, it is based on the parent list accumulated in the deletion information storage part 1040, that is, deletion information, the information (for example, information which shows a child node) about the child node registered into the child list of a parent node to this parent node registered into the parent list

されている親リストつまり削除情報に基づいて、ノードNについて、親リストに登録されている親ノードから、該親ノードの子リストに登録されている子ノードに関する情報（例えば子ノードを示す情報）を取り除く。この処理が終了した後は、ノードNには出力すべきノードのみが存在していることになる。

【0120】

そしてステップ1072を終了した後、出力処理部103は、ノードNについての子リストに含まれない子をノード（つまり出力すべきノード）に対して、再帰的にノード出力処理を実行する（ステップ1073）。

【0121】

一方、ステップ1071において未登録の場合は、ノードNをそのまま出力し（ステップ1074）、そのノードNの子ノード全てに対して、再帰的にノード出力処理を実行する（ステップ1075）。

【0122】

以上説明したように第3の実施例によれば、パターン照合して削除されるノード（ノードC）が検出されると、このノードCと該ノードCの親のノード（ノードP）とを対応して記憶し、

about Node N is removed.

After this processing is completed, only the node which should be outputted exists in Node N.

[0120]

And after completing step 1072, the output-processing section 103 performs node output processing for the child who is not contained in the child list about Node N recursively to a node (node which should be outputted in other words) (step 1073).

[0121]

On the other hand, in step 1071, in not registering, Node N is outputted as it is (step 1074), and it performs node output processing recursively to all the child nodes of the node N (step 1075).

[0122]

According to 3rd Example as explained above, if it detects the node (node C) deleted by carrying out a pattern match, it corresponds and this node C and the node (node P) of the parents of this node C are stored, and the information (information which shows Node C)

そして、親のノードPからノードCに関する情報（ノードCを示す情報）を取り除いて、その親のノードPを出力するようにしているので、ユーザが、親のノードPの内容を変更する必要がない。このため構造化文書に対する編集処理の作業効率を向上させることができる。

about Node C is removed from parents' node P, the parents' node P is outputted.

Therefore, a user does not need to alter the content of parents' node P.

For this reason, the working efficiency of the edit processing with respect to a structuring document can be improved.

【0123】

次に第4の実施例を図21乃至図27を参照して説明する。

[0123]

Next, 4th Example is demonstrated with reference to FIG. 21 - FIG. 27.

【0124】

図21は本発明に係る文書処理装置の第4の実施例を示す機能ブロック図である。同図において、文書処理装置は、メモリ1610、解釈部1620、再編成部1630、ソース文書ファイル群1640、ターゲット文書ファイル群1650、照合部1660、複数抽出点蓄積部1670、複数挿入点蓄積部1680、出力処理部1690、抽出挿入関係ルール指定部1700を備えている。

[0124]

FIG. 21 is a functional-block figure which shows 4th Example of the word-processing apparatus based on this invention.

In this figure, the word-processing apparatus is equipped with memory 1610, the interpretation section 1620, the reorganization section 1630, the source text-file group 1640, the target text-file group 1650, the collation section 1660, two or more extracting point storage part 1670, two or more inserting point storage part 1680, the output-processing section 1690, and the extraction insertion relationship rule designation section 1700.

【0125】

メモリ1610には、第1の実施例で説明したようなパターン記述情報1611が記憶されており、解釈部1620は、メモリ1610からパターン記述情報1611を読み出して解釈

[0125]

The pattern description information 1611 that it demonstrated in 1st Example is stored in memory 1610, the interpretation section 1620 reads and interprets the pattern description information 1611 from memory 1610, this interpretation result is stored in memory 1610

し、この解釈結果を第1の実施例で説明したような文書構造パターン1612としてメモリ1610に記憶する。

【0126】

再編成部1630は、ソース文書ファイル群1640に保持されている複数の構造化文書内を走査して、これらの構造化文書を照合処理可能な形式の構造に再編成し、この結果をソース再編成構造化文書群1613としてメモリ1610に格納する。同様に、ターゲット文書ファイル群1650内の複数の構造化文書についても、ソース再編成構造化文書群1614としてメモリ1610に格納する。

【0127】

この第4の実施例において、構造化文書とは、章、節といった文書構造と文書内容とを一緒に持つフォーマットによる文書表現を意味している。1つのファイル内に複数の構造化文書を含むとは、図22に示す様に、1つのファイル内に、異なる論理根を持つ論理構造が複数含まれることを示している。論理根が異なる論理構造は互いに独立であり、且つ部分構造が共有されることはない。部分構造とは、構造化文書における一部分の構造のことであり、例えば図22中

as a document structure pattern 1612 which was demonstrated in 1st Example.

[0126]

The reorganization section 1630 scans the inside of two or more structuring documents currently maintained at the source text-file group 1640, these structuring documents are reorganized in the structure of the form which can carry out collation processing, it stores in memory 1610 by making this result into the source reorganization structuring document group 1613.

Similarly, it stores in memory 1610 as a source reorganization structuring document group 1614 also about two or more structuring documents in the target text-file group 1650.

[0127]

In this 4th Example, the structuring document is meaning the document expression by the format which has document structure, such as a chapter and a clause, and the content of a document together.

Two or more structuring documents are included in one file, as shown in FIG. 22, it is showing that two or more logic structure which has a different logic solution in one file is included.

The logic structure from which a logic solution differs is mutually independent.

And the partial structure is not shared.

Partial structure is a part of structure in a structuring document.

点線で囲まれた構造化文書においては、「節」というノード以下の構造や、「表題」というノード以下の構造などである。

For example, in the structuring document surrounded by the dotted line, they are the structure below the node a "clause", the structure below the node a "title", etc., during FIG. 22.

【0128】

ソース文書ファイルとは、部分構造を他の文書へ挿入するために、文書構造パターンに適合する部分構造が抽出される文書ファイル（つまりソース側の文書ファイル）のことである。結果としてこの文書ファイルの内容は変更されることはない。

[0128]

A source text file is a text file (in other words text file by the side of a source) with which the partial structure which adapts a document structure pattern is extracted, in order to insert the partial structure in other documents.

The content of this text file is not altered as a result.

【0129】

一方、ターゲット文書ファイルとは、他の文書からの部分構造を挿入するために、文書構造パターンに適合する部分構造が抽出される文書ファイル（つまりターゲット側の文書ファイル）のことである。結果としてこの文書ファイルの内容は変更される。

[0129]

On the other hand, a target text file is a text file (in other words text file by the side of a target) with which the partial structure which adapts a document structure pattern is extracted, in order to insert the partial structure from other documents.

The content of this text file is altered as a result.

【0130】

またソース再編成構造化文書とは、ソース文書ファイル内の構造化文書に対する再編成処理の結果である再編成構造化文書のことである。

[0130]

Moreover, a source reorganization structuring document is a reorganization structuring document which is the result of the reorganization processing with respect to the structuring document in a source text file.

【0131】

一方、ソース再編成構造化文書

[0131]

On the other hand, a source reorganization

とは、ターゲット文書ファイル内の構造化文書に対する再編成処理の結果である再編成構造化文書のことである。

structuring document is a reorganization structuring document which is the result of the reorganization processing with respect to the structuring document in a target text file.

【0132】

照合部1660は、複数ファイル抽出点認識部1661と、複数ファイル挿入点認識部1662とを有している。複数ファイル抽出点認識部1661は、ソース構造化文書群をそれぞれ格納する複数のファイルに対して、文書構造パターンによる照合によりファイルを走査し、複数の部分構造の抽出点を認識し、この認識結果を複数抽出点蓄積部1670に格納する。このときファイル名と抽出点の対の情報を格納する。一方、複数ファイル挿入点認識部1662は、ターゲット構造化文書群をそれぞれ格納する複数ファイルに対し、文書構造パターンによりファイルを走査し、複数の部分構造に対する挿入点を認識すると共に、この認識結果を複数挿入点蓄積部1680に格納する。このときファイル名と挿入点の対の情報を格納する。なお複数抽出点蓄積部1670と複数挿入点蓄積部1680とは独立しているが、抽出点の情報と挿入点の情報とを区別するようにして、これらの情報を1つの蓄積部に蓄積するようにしても

[0132]

The collation section 1660 has the two or more files extracting point recognition part 1661 and two or more files inserting point recognition part 1662.

Two or more files extracting point recognition part 1661 scans a file by collation by a document structure pattern to two or more files which store a source structuring document group, respectively, the extracting point of view of two or more partial structure is recognized, this recognition result is stored in the two or more extracting point storage part 1670.

At this time, a pair of information on a file name and an extracting point of view is stored.

On the other hand, two or more files inserting point recognition part 1662 scans a file with a document structure pattern to two or more files which store a target structuring document group, respectively, while recognizing the inserting point of view with respect to two or more partial structure, this recognition result is stored in the two or more inserting point-of-view storage part 1680.

At this time, a pair of information on a file name and an inserting point of view is stored.

In addition, two or more extracting point storage part 1670 and two or more inserting point storage part 1670 are independent.

However, as the information on an extracting point and the information on an inserting point

良い。

are distinguished, it may make it accumulate such information in one storage part.

【0133】

抽出挿入関係ルール指定部1700は、抽出点と挿入点との対応関係を、ファイルを跨がる1対1、あるいはファイルを跨がる複数n対1のいずれかのルールを出力処理部1690に与える。

[0133]

The extraction insertion relationship rule designation section 1700 gives the rule in any one of 1 to 1 which straddles a file, or few n to 1 file which straddles a file to the output-processing section 1690 for the correspondence of an extracting point and an inserting point.

【0134】

出力処理部1690には、複数ファイル部分構造抽出挿入部1691が設けられており、複数ファイル部分構造抽出挿入部1661は、抽出挿入関係ルール指定部1700から与えられた抽出点と挿入点との対応関係の情報に基づいて、複数抽出点蓄積部1670に蓄積されている抽出点に対応する部分構造から、複数挿入点蓄積部1680に蓄積されている挿入点に対応する部分構造分への文書構造の挿入操作を実行する。

[0134]

Two or more files partial structure extraction insertion part 1691 is provided in the output-processing section 1690, the file partial structure extraction insertion part 1661 is two or more, it is based on the information on the correspondence of the extracting point and the inserting point which it imparted from the extraction insertion relationship rule designation section 1700, insertion operation of the document structure from the partial structure corresponding to the extracting point accumulated in the extracting point storage part 1670 to a part for the partial structure corresponding to the inserting point accumulated in the inserting [two or more] point storage part 1680 is performed. [two or more]

【0135】

この図21に示した装置も、図2に示した第1の実施例のハードウェア構成で実現することができる。ここで、図21に示し

[0135]

The apparatus shown in this FIG. 21 is also realizable with the hardware composition of 1st Example shown in FIG. 2. Here, the correspondence of the component of

た機能ブロック図の構成要素と図2に示したブロック図の構成要素との対応関係について説明する。図26に示したメモリ1610は図2に示した主メモリ20に対応し、図21に示した解釈部1620、再編成部1630、照合部1660、出力処理部1690及び抽出挿入関係ルール指定部1700は共に図2に示したCPU210に対応し、ソース文書ファイル群1640及びターゲット文書ファイル群1650は共に図2に示したディスク230に対応している。

【0136】

この第4の実施例も、基本的には第1の実施例と同様である。第1の実施例と異なるのは、1つのファイル内の複数の構造化文書に対して、文書構造パターンに一致する構造を抽出する点である。また複数の構造化文書を有するファイルを複数設け、これらのファイル内の複数の構造化文書に対して照合する点も異なっている。

【0137】

そこで、第4の実施例における文書編集処理について、図23乃至図27を参照して説明する。

the functional-block figure shown in FIG. 21 and the component of a block diagram shown in FIG. 2 is demonstrated.

Memory 1610 shown in FIG. 26 corresponds to the main memory 20 shown in FIG. 2, both the interpretation section 1620 shown in FIG. 21, the reorganization section 1630, the collation section 1660, the output-processing section 1690, and the extraction insertion relationship rule designation section 1700 correspond to CPU210 shown in FIG. 2, both the source text-file group 1640 and the target text-file group 1650 are corresponding to disc 230 shown in FIG. 2.

[0136]

This 4th Example is the same as 1st Example fundamentally.

Differing from 1st Example is the point of extracting the structure which is in agreement with a document structure pattern to two or more structuring documents in one file.

Moreover, two or more files which have two or more structuring documents are provided, and the points of view compared to two or more structuring documents in these files also differ.

[0137]

Then, the text editing processing in 4th Example is demonstrated with reference to FIG. 23 - FIG. 27.

【0138】

パターン記述情報20として、
 (節／表題／まとめ) # 本
 文段落
 (3)
 ここで、／は包含関係を示す記
 号
 #は順序関係を示す記号
 が記述されメモリ1610に記
 憶されている。

[0138]

As the pattern description information 20,
 (Clause / title / conclusion) # text stage... (3)
 Here, / is a symbol which shows an inclusive
 relation.
 # is the symbol which shows order relation
 These are described and it stores in memory
 1610.

【0139】

次に解釈部1630によって、
 図4に示す第1の実施例のパタ
 ーン解釈処理手順が実行される
 ことにより上記(3)のパター
 ン記述情報が解釈され、更にこ
 の結果が文書構造パターン16
 12としてメモリ1610に記
 憶される。

[0139]

Next, when pattern interpretation procedure of
 1st Example shown in FIG. 4 is performed by
 the interpretation section 1630, the pattern
 description information on said (3) is
 interpreted, furthermore, this result is stored in
 memory 1610 as a document structure pattern
 1612.

【0140】

続いて再編成部1930によっ
 て、ソース文書ファイル群16
 40とターゲット文書ファイル
 群1650とが再編成され、更
 にこれらの結果が、ソース再編
 成構造化文書群1613、ター
 ゲット再編成構造化文書群16
 14としてメモリ1610に記
 憶される。

[0140]

Then, the source text-file group 1640 and the
 target text-file group 1650 are reorganized by
 the reorganization section 1930, furthermore,
 these results are stored in memory 1610 as the
 source reorganization structuring document
 group 1613 and a target reorganization
 structuring document group 1614.

【0141】

続いて照合部1660の複数フ
 ァイル抽出点認識部1661に
 よる抽出点認識処理について、

[0141]

Then, the extracting point-of-view recognition
 processing by the extracting [two or more files]
 point-of-view recognition part 1661 of the

図 23 を参照して説明する。図 23 はその処理動作を示すフローチャートである。

【0142】

複数ファイル抽出点認識部 161 は、最初のソース文書ファイル（ソース再編成構造化文書群 1613 中の 1 つのファイル）をメモリ 1610 から読み込んで（ステップ 1801）、ソース文書ファイルは終りか否かを判断し（ステップ 1802）、終りの場合には処理を終了し、一方、終りでない場合は、ファイル内の全ての構造化文書（つまり論理根を持つ文書）に対する処理が終了したか否かを判断する（ステップ 1803）。

【0143】

ここで、まだ未処理の構造化文書が存在している場合は、その構造化文書に対するパターン照合処理を実行し（ステップ 1804）、その照合処理結果である抽出点を複数抽出点蓄積部 1670 に蓄積する（ステップ 1805）。

【0144】

上記ステップ 1803 において、全ての構造化文書について処理した場合は、次のソース文書ファイルをメモリ 1610 か

collation section 1660 is demonstrated with reference to FIG. 23.

FIG. 23 is a flowchart which shows the processing operation.

[0142]

The extracting [file] point-of-view recognition part 1661 reads the first source text file (one file in the source reorganization structuring document group 1613) from memory 1610 (step 1801), and a source text file judges 可否 finally (step 1802), and, in the case of the end, completes two or more processing, on the other hand, when it finally is not, it is judged whether the processing with respect to all the structuring documents in a file (document which has a logic solution in other words) was completed (step 1803).

[0143]

Here, when the untreated structuring document still exists, pattern-match processing with respect to the structuring document is performed (step 1804), and the extracting point of view which is the collation processing result is accumulated in the extracting [two or more] point-of-view storage part 1670 (step 1805).

[0144]

In the above-mentioned step 1803, when all structuring documents are processed, the next source text file is read from memory 1610, and, after that, it returns to the above-mentioned step

ら読み込み、その後、上記ステップ1802に戻る。

【0145】
なおステップ1804のパターン照合処理は、図8に示す第1の実施例の処理手順と同様である。

【0146】
同様にして、複数ファイル挿入点認識部1662は、ターゲット文書ファイル（ターゲット再編成構造化文書群1614）に対する挿入点の認識処理を実行する。この結果は、複数挿入点蓄積部1680に蓄積される。

【0147】
すなわち、複数ファイル抽出点認識部1661と複数ファイル挿入点認識部1662は基本的には同様の処理を実行し、異なるのは、対象となる文書ファイル（構造化文書）がソースであるかターゲットであるかという点である。

【0148】
ここで、抽出点の認識処理結果の様子を図24に示す。図24において、ファイル1、ファイル2は、ソース再編成構造化文書を示しており、またハッチングの掛った部分が、文書構造パターン1612に適合した部分

1802.

[0145]
In addition, pattern-match processing of step 1804 is the same as that of the procedure of 1st Example shown in FIG. 8.

[0146]
Similarly, two or more files inserting point recognition part 1662 performs recognition processing of the inserting point with respect to a target text file (target reorganization structuring document group 1614). This result is accumulated in two or more inserting point storage part 1680.

[0147]
That is, two or more files extracting point recognition part 1661 and the two or more files inserting point recognition part 1662 perform similar processing fundamentally, differing is the point whether the text file (structuring document) used as object is a source, or to be a target.

[0148]
Here, the state of the recognition processing result of an extracting point of view is shown in FIG. 24.
In FIG. 24, file 1 and file 2 are showing the source reorganization structuring document, moreover, the part which hatching required is a part which adapted the document structure

である。この図 2 4 から分かるように、ファイル内の複数の構造化文書及び複数のファイルに跨がって、構造がパターンマッチングされ適合されている。この例での抽出点は、ハッチングの掛った部分の「節」というノードの直前の位置（つまり「論理根」というノードとの接続点の位置）である。この抽出点は、各ファイル毎に抽出点の列として複数抽出点蓄積部 1 6 7 0 に蓄積される。

【0149】

同様に挿入点の認識処理結果も、図 2 4 に示す様に、文書構造パターン 1 6 1 2 に適合した部分が認識されることとなる。挿入点についても上記同様に考えることができる。

【0150】

以上の説明から分かるように、この第 4 の実施例においては、図 2 4 に示すように、文書構造パターン 1 6 1 2 に適合する部分構造（ハッチング部分）を抽出することが、本来の目的ではなく、「節」というノードを抽出することが目的なのである。しかし、図 2 4 に示されるように、「節」というノード以下の構造には各種の部分構造が接続されているので、所望の「節」とい

pattern 1612.

Ranging over two or more structuring document and two or more files in a file, the pattern matching of the structure is carried out and it adapts so that this FIG. 24 may show.

The extracting point in this example, is a position in front of the node the "clause" of the part which required hatching (position of a connecting point with the node, in other words a "logic solution").

This extracting point of view is accumulated in two or more extracting point-of-view storage part 1670 as a row of an extracting point of view for every file.

[0149]

Similarly, as the recognition processing result of an inserting point of view is also shown in FIG. 24, the part which adapted the document structure pattern 1612 will be recognized.

An inserting point can also be considered in the same manner to the above.

[0150]

In this 4th Example so that the above explanation may show, show in FIG. 24, it is not the original objective to extract the partial structure (hatching part) which adapts the document structure pattern 1612, it is the objective to extract the node a "clause."

However, as FIG. 24 shows, various kinds of partial structure is connected to the structure below the node a "clause."

Therefore, in order to extract the node desired "node", collation with the document structure pattern 1612 is implemented.

うノードを抽出するために、文書構造パターン1612との照合を実施しているのである。

【0151】

次に、出力処理部1690の複数ファイル部分構造抽出挿入部1691の出力処理について、図25を参照して説明する。図25はその処理動作を示すフローチャートである。

[0151]

Next, output processing of two or more files partial structure extraction insertion part 1691 of the output-processing section 1690 is demonstrated with reference to FIG. 25.

FIG. 25 is a flowchart which shows the processing operation.

【0152】

複数ファイル部分構造抽出挿入部1691は、複数抽出点1蓄積部1670から各ファイル毎の抽出点の列を得る。これらを $A[i] = (\text{file 名, 抽出点})$ に順に格納すると共に (ステップ2001)、複数挿入点蓄積部1680から各ファイル毎の挿入点の列を得る。これらを $B[j] = (\text{file 名, 挿入点})$ に順に格納する (ステップ2002)。

[0152]

The file partial structure extraction insertion part 1691 obtains two or more rows of the extracting point for every file from two or more extracting point 1 storage part 1670.

While storing these in $A[i] = (\text{file name, extracting point})$ in order (step 2001), the row of the inserting point for every file is obtained from two or more inserting point storage part 1680.

These are stored in $B[j] = (\text{file name, inserting point})$ in order (step 2002).

【0153】

次に、挿入抽出関係ルール指定部1700から指定された抽出点と挿入点との対応関係のルールが“ファイルを跨がる1対1”であるか否かを判断する (ステップ2003)。

[0153]

Next, it is judged whether the rule of the correspondence of the extracting point of view and the inserting point of view which were designated from the insertion extraction relationship rule designation section 1700 is "1 to 1 which straddles a file" (step 2003).

【0154】

ここで、“ファイルを跨がる1対1”の場合は、 $i = 1$ 、 $j = 1$

[0154]

Here, in the case of "1 to 1 which straddles a file", it defines as $i = 1$ and $j = 1$ (step 2004), it is

と定義し (ステップ 2 0 0 4)、judged whether A [i] or B [j] is end (step 2005).
A [i] あるいは B [j] が終
りか否かを判断する (ステップ
2 0 0 5)。

【0155】

ここで、終りでない場合は、A
[i] に示される抽出点に基づ
いて、ソース文書ファイル群か
ら部分構造を抽出すると共に
(ステップ 2 0 0 6)、この部分
構造を B [j] に示される挿入
点に挿入する (ステップ 2 0 0
7)。

[0155]

When it is not end here, while extracting the
partial structure from a source text-file group
based on the extracting point shown by A [i]
(step 2006), this partial structure is inserted in
the inserting point shown by B [j] (step 2007).

【0156】

その後、 $i = i + 1$ 、 $j = j +$
1 と再定義した後 (ステップ 2
0 0 8)、上記ステップ 2 0 0 5
に戻る。ステップ 2 0 0 5 にお
いて A [i] あるいは B [j]
が終りの場合は、結果を出力す
る (ステップ 2 0 0 9)。

[0156]

After that, after redefining it as $i = i + 1$ and $j = j + 1$
(step 2008), it returns to the above-mentioned
step 2005.
A result is outputted when A [i] or B [j] is end in
step 2005 (step 2009).

【0157】

上記ステップ 2 0 0 3 において
ルールが “ファイルを跨がる 1
対 1” でない場合は、ルールが
“ファイルを跨がる複数 n 対
1” であるか否かを判断する (ス
テップ 2 0 1 0)。そうであれ
ば、 $j = 1$ と定義し (ステップ
2 0 1 1)、その後、B [j] が
終りであるか否かを判断する
(ステップ 2 0 1 2)。

[0157]

When a rule is not "1 to 1 which straddles a file"
in the above-mentioned step 2003, a rule
judges whether it is two or more-n to 1" which
straddles "file (step 2010).
If that is right, it will define as $j = 1$ (step 2011),
and, after that, it will be judged whether B [j] is
end (step 2012).

【0158】

ここで、終りの場合は、A[1]～A[n]に示される抽出点に基づいて、ソース文書ファイル群から部分構造を全て抽出し、これらA[1]～A[n]までの部分構造を兄弟として繋ぐと共に(ステップ2013)、兄弟として繋がれた構造を、B[j]に示される挿入点に挿入する(ステップ2014)。この挿入点に対して、兄、弟、子供として挿入することができる。

[0158]

Here, in the case of the end, based on the extracting point of view shown by A[1]-A[n], all partial structure is extracted from a source text-file group, while connecting the partial structure to these A[1]-A[n] as a brother (step 2013), the structure connected as a brother is inserted in the inserting point shown by B[j] (step 2014).

It can insert as an elder brother, a younger brother, and a child to this inserting point of view.

【0159】

上記ステップ2014を終了した後は $j = j + 1$ と再定義し(ステップ2015)、その後、上記ステップ2012に戻る。すなわち結果として、B[1]～B[n]の各挿入点に、兄弟として繋がれたA[1]～A[n]までの部分構造が挿入される。

[0159]

After completing the above-mentioned step 2014, it is redefined as $j=j+1$ (step 2015), and after that, it returns to the above-mentioned step 2012.

That is, as a result, the partial structure to A[1]-A[n] connected with each inserting point of view of B[1]-B[n] as a brother is inserted.

【0160】

上記ステップ2012においてB[j]が終了した場合は上記ステップ2009に進む。

[0160]

When B[j] is completed in the above-mentioned step 2012, it progresses to the above-mentioned step 2009.

【0161】

上記ステップ2010においてルールが“ファイルを跨がる複数n対1”でない場合は挿入処理は行わない(ステップ2016)。

[0161]

Insertion processing is not performed when a rule is not two or more-n to 1" which straddles "file in the above-mentioned step 2010 (step 2016).

【0162】

なお複数ファイル部分構造抽出挿入部1691は、ソース文書ファイル群の抽出点、ターゲット文書ファイル群の挿入点のいずれかの数が多いときは挿入処理を行わず、ステータスを返す。

[0162]

In addition, two or more files partial structure extraction insertion part 1691 does not process insertion, when there are many numbers in any one of the extracting point of view of a source text-file group and the inserting point of view of a target text-file group, but it returns the status.

【0163】

例えば、抽出点の数>挿入点の数、のときステータスの値が1
抽出点の数<挿入点の数、のときステータスの値が2
この結果として、出力処理部1690からは、エラー通知が出力される。

[0163]

For example, at the time of number of extracting points >

The number of inserting points, the value of the status is 1.

At the time of several extracting point, <

The number of inserting points, the value of the status is 2 at these times.

As this result, an error notification is outputted from the output-processing section 1690.

【0164】

また、抽出点の数=挿入点の数、のときは0のステータスを返す。この結果として、出力処理部1690からは、変更後のターゲット文書ファイルが出力されることとなる。

[0164]

Moreover, the status of 0 is returned at the time of the number of extracting point = number of an inserting point.

As this result, the target text file after alteration will be outputted from the output-processing section 1690.

【0165】

ここで、ファイルを跨がる1対1のルールに基づく挿入結果の様子を図21に示し、またファイルを跨がる複数n対1のルールに基づく挿入結果の様子を図22に示す。

[0165]

Here, the state of the insertion result based on the rule of 1 to 1 which straddles a file is shown in FIG. 21, moreover, the state of the insertion result based on [two or more] the rule of n to 1 which straddles a file is shown in FIG. 22.

【0166】**[0166]**

以上説明したように第4の実施例によれば、複数の文書ファイルであって、且つ1つのファイル中に複数の構造化文書文書が保存されている場合であっても、構造化文書内のオブジェクト間の関係を利用したパターン、つまり基準となる階層構造のパターンとのパターンマッチングを行うようにしているので、構造化文書における正確な情報（文書構成要素）にアクセスすることができると共に、階層上のデータ（文書構成要素）の位置を簡単に指定することができる。

[0167]

すなわち、ターゲット側の複数のファイルそれぞれに保存されている複数の構造化文書から抽出された文書構成要素に対する、ソース側の複数のファイルそれぞれに保存されている複数の構造化文書から抽出された文書構成要素の挿入操作を一度に自動的に行うことができるということである。

[0168]

次に第5の実施例を図28乃至図31を参照して説明する。

[0169]

図28は本発明に係る文書処理装置の第5の実施例を示す機能

As explained above, according to 4th Example, they are two or more text files, comprised such that and even if it is the case where two or more structuring document documents are conserved in one file, it is made to perform the pattern matching of the pattern using the relationship between the objects in a structuring document, and the pattern of the hierarchical structure which constitutes a reference standard in other words.

Therefore, while being able to access to the exact information (document component) in a structuring document, the position of the data on a hierarchy (document component) can be designated easily.

[0167]

That is, insertion of the document component extracted from two or more structuring documents with respect to the document component extracted from two or more structuring documents conserved at two or more files of each by the side of a target conserved at two or more files of each by the side of a source can be operated automatically at once.

[0168]

Next, 5th Example is demonstrated with reference to FIG. 28 - FIG. 31.

[0169]

FIG. 28 is a functional-block figure which shows 5th Example of the word-processing apparatus

ブロック図である。この機能ブロック図は、図 1 に示した第 1 の実施例の機能ブロック図の構成において、ファイル位置情報保持部 2 3 1 0、属性情報指定部 2 3 2 0 を追加し、出力処理部 9 0 を出力処理部 2 3 3 0 に変更した構成になっている。なお図 2 3 において、図 1 に示した構成要素と同様の機能を果たす部分には同一の符号を付している。

【0170】

ファイル位置情報保持部 2 3 1 0 は、照合部 8 0 の照合結果である文書構成要素のファイル内の位置情報を保持する。

【0171】

属性情報指定部 2 3 2 0 は、文書構成要素の属性情報を指定するものであり、属性値の参照のときは属性名を指定し、属性値の変更のときは属性名及び属性値を指定する。

【0172】

出力処理部 2 3 3 0 は、ファイル位置情報保持部 2 3 1 0 に保持されている文書構成要素のファイル内の位置情報と、属性情報指定部 2 3 2 0 から指定される属性情報とに基づいて出力処

based on this invention.

In the composition of the functional-block figure of 1st Example which this functional-block figure showed in FIG. 1, it has the composition of having added the file positional-information retainer 2310 and the attribute information designation section 2320, and having altered the output-processing section 90 into the output-processing section 2330.

In addition, in FIG. 23, the same code is attached into the part which achieves the function similar to the component shown in FIG. 1.

[0170]

The file positional-information retainer 2310 maintains the positional information in the file of the document component which is the collation result of the collation section 80.

[0171]

The attribute information designation section 2320 designates the attribute information on a document component.

An attribute name is designated at the time of refer of attribute value, an attribute name and attribute value are designated at the time of alteration of attribute value.

[0172]

The output-processing section 2330 implements output processing based on the positional information in the file of the document component currently maintained at the file positional-information retainer 2310, and the attribute information designated from the

理を実施する。ここで、属性値の参照のときは、該当する文書構成要素の属性名を持つ属性の属性値を出力し、一方、属性値の変更のときは、該当する文書構成要素の属性名を持つ属性を、指定された属性値に変更して出力する。

attribute information designation section 2320.

The attribute value of an attribute with the attribute name of the corresponding document component is outputted at the time of refer here of attribute value, on the other hand, an attribute with the attribute name of the corresponding document component is altered and outputted to the designated attribute value at the time of alteration of attribute value.

【0173】

なおここでは構造化文書は、図29(a)に示す様に各文書構成要素をノードとする木構造を持っているが、ファイル上では、図29(b)に示す様に、決められた規則に従って各文書構成要素は1列に並んでいる。また文書構成要素内の属性名と属性値の対も1列に並んでいるとする。

[0173]

In addition, the structuring document has the tree structure which makes each document component a node as shown in FIG.29(a) here. However, on the file, as shown in FIG.29(b), each document component is located in a line with one row according to the decided rule.

Moreover, suppose that the pair of the attribute name and attribute value in a document component is also ranked with one row.

【0174】

図28に示した装置も、図2に示した第1の実施例のハードウェア構成で実現することができる。ここで図28に示した機能ブロック図の構成要素と図2に示したブロック図の構成要素との対応関係について説明する。図28に示したファイル位置情報保持部2310は図2に示した主メモリ20に対応し、図28に示した属性指定部2320及び出力処理部2330は共に図2に示したCPU210に対

[0174]

The apparatus shown in FIG. 28 is also realizable with the hardware composition of 1st Example shown in FIG. 2.

The correspondence of the component of the functional-block figure shown in FIG. 28 here and the component of a block diagram shown in FIG. 2 is demonstrated.

The file positional-information retainer 2310 shown in FIG. 28 corresponds to the main memory 20 shown in FIG. 2, both the attribute designation sections 2320 and output-processing sections 2330 that were shown in FIG. 28 are corresponding to CPU210

応じている。他の構成要素については第1の実施例と同様である。

【0175】

この第5の実施例も、基本的には第1の実施例と同様である。第1の実施例と異なるのは、構造化文書中から、文書構造パターンに一致する構造を抽出し、この抽出した構造に対して、属性の参照又は変更の処理を施すという点である。

【0176】

なおこの第5の実施例においては、照合部8.0による文書構造パターンと再編成構造化文書との照合処理までは、第1の実施例で説明した処理と同様なので、ここではその説明を省略し、属性の参照又は変更処理について説明する。

【0177】

次に、文書処理装置の属性の参照又は変更処理について、図3.0及び図3.1を参照して説明する。図3.0は属性の参照処理動作のフローチャートを示し、図3.1は属性の変更処理動作のフローチャートを示している。

【0178】

最初に属性の参照処理について

shown in FIG. 2.

About other components, it is the same as that of 1st Example.

[0175]

This 5th Example is the same as 1st Example fundamentally.

Differing from 1st Example is the point of extracting the structure which is in agreement with a document structure pattern, and performing refer of an attribute, or processing of alteration to this extracted structure out of a structuring document.

[0176]

In addition, since collation processing with the document structure pattern and reorganization structuring document by the collation section 80 is the same as the processing demonstrated in 1st Example in this 5th Example, here, the explanation is omitted and refer of an attribute or alteration processing is demonstrated.

[0177]

Next, refer or alteration processing of the attribute of a word-processing apparatus is demonstrated with reference to FIG.30 and FIG.31.

FIG. 30 shows the flowchart of the refer processing operation of an attribute, FIG. 31 is showing the flowchart of the alteration processing operation of an attribute.

[0178]

Refer processing of an attribute is

説明する。図30に示すように、出力処理部2330は、ファイル位置情報保持部2310から属性結果（ファイル位置）を1つ取り出し（ステップ2501）、文書ファイルの「読み出し位置」を読み出した照合結果に設定すると共に（ステップ2502）、その読み出し位置に存在する文書構成要素内から属性を1つ読み込む（ステップ2503）。

[0179]

次に出力処理部2330は、その属性名は属性情報指定部2320から指定された属性名と同じであるか否かを判断する（ステップ2504）。

[0180]

ここで、同一の場合はその属性値を出力用のファイルに書き出し（ステップ2505）、その後、属性は終りか否かを判断する（ステップ2506）。

[0181]

ここで、終りでない場合は、上記ステップ2503に戻りこのステップ以降を実行する。すなわち1つの文書構成要素内に存在する全ての属性についての属性の参照の処理を実施する。

[0182]

demonstrated initially.

As shown in FIG. 30, the output-processing section 2330 takes out one attribute result (file position) from the file positional-information retainer 2310 (step 2501), while setting the "read-out position" of a text file as read collation result (step 2502), one attribute is read from the inside of the document component which exists in the read-out position (step 2503).

[0179]

Next, the output-processing section 2330, judged whether the attribute name is the same as the attribute name designated from the attribute information designation section 2320 (step 2504).

[0180]

Here, when the same, the attribute value is written out to the file for an output (step 2505), and after that, an attribute judges whether it is end (step 2506).

[0181]

Here, when it is not end, it returns to the above-mentioned step 2503, and this step or subsequent ones is performed.

That is, refer of the attribute about all the attributes that exist in one document component are processed.

[0182]

ステップ2506において属性が終りの場合は、照合結果は終りか、つまりファイル位置情報保持部2310に保持されている全ての照合結果について処理したか否かを判断する（ステップ2507）。

When an attribute is end in step 2506, it is judged whether the collation result was processed about all the collation results currently maintained at end, in other words the file positional-information retainer 2310 (step 2507).

【0183】

ここで、未処理の照合結果がある場合には上記ステップ2501に戻りこのステップ以降を実行し、一方、全て処理した場合は属性の参照処理を終了する。

[0183]

Here, when there is a untreated collation result, it returns to the above-mentioned step 2501, and this step or subsequent ones is performed, on the other hand, when all are processed, refer processing of an attribute is completed.

【0184】

なおステップ2504において指定された属性名でない場合は何もしないでステップ2506に進む。

[0184]

In addition, it progresses to step 2506 without doing anything, when it is not the attribute name designated in step 2504.

【0185】

次に属性の変更処理について説明する。図3-1に示す様に、出力処理部2330は、文書ファイルの「読み出し位置」を先頭に設定すると共に（ステップ2601）、ファイル位置情報保持部2310から照合結果（ファイル位置）を1つ取り出す（ステップ2602）。次に、文書ファイルにおける照合結果の位置までの部分をそのまま出力用のファイルに書き出す（ステップ2603）。

[0185]

Next, alteration processing of an attribute is demonstrated.

As shown in FIG. 31, while the output-processing section 2330 sets the "read-out position" of a text file as a head (step 2601), one collation result (file position) is taken out from the file positional-information retainer 2310 (step 2602).

Next, the part to the position of the collation result in a text file is written out to the file for an output as it is (step 2603).

【0186】

続いて、文書ファイルの「読み出し位置」を照合結果に設定すると共に（ステップ2604）、その読み出し位置に存在する文書構成要素内から属性を1つ読み込む（ステップ2605）。

[0186]

Then, while setting the "read-out position" of a text file as a collation result (step 2604), one attribute is read from the inside of the document component which exists in the read-out position (step 2605).

【0187】

そして、その属性名は属性情報指定部2320から指定された属性名と同じであるか否かを判断する（ステップ2606）。ここで、同一の場合は属性名と指定された属性値とを出力用のファイルに書き出し（ステップ2607）、同一でない場合は、属性名と読み込んだ属性値とを出力用のファイルに書き出す（ステップ2608）。

[0187]

And it is judged whether the attribute name is the same as the attribute name designated from the attribute information designation section 2320 (step 2606).

Here, when the same, the attribute value designated as the attribute name is written out to the file for an output (step 2607), when not the same, an attribute name and the read attribute value are written out to the file for an output (step 2608).

【0188】

ステップ2607あるいはステップ2608を終了したら、属性は終りか否かを判断する（ステップ2609）。ここで、終りでない場合は、上記ステップ2605に戻りこのステップ以降を実行する。すなわち1つの文書構成要素内に存在する全ての属性について属性の変更処理を実施する。

[0188]

If step 2607 or step 2608 is completed, it will judge that an attribute is about whether it is end (step 2609).

Here, when it finally is not, it returns to the above-mentioned step 2605, and this step or subsequent ones is performed.

That is, alteration of an attribute is processed about all the attributes that exist in one document component.

【0189】

ステップ2609において属性が終りの場合は、照合結果は終

[0189]

When an attribute is finally in step 2609, as for a collation result, having processed all the

りか、つまりファイル位置情報保持部2310に保持されている全ての照合結果について処理したか否かを判断する（ステップ2610）。

【0190】

ここで、未処理の照合結果がある場合には上記ステップ2602に戻りこのステップ以降を実行し、一方、全て処理した場合は、文書ファイルの最後までをそのまま出力用のファイルに書き出す（ステップ2611）。

【0191】

この第5の実施例においては、属性の参照により取り出された属性は、外部の汎用的な演算手段を用いることにより加工することができる。その加工結果を属性情報指定部2320から属性情報として設定することにより、文書内に付加することができる。

【0192】

以上説明したように第5の実施例によれば、パターン記述情報と、属性名か属性名及び属性値を設定することにより、構造化文書内のオブジェクト間の関係を利用したパターン、つまり基準となる階層構造のパターンとのパターンマッチングを実施し、マッチした部分（文書構成

collation results currently maintained at the file positional-information retainer 2310 judges the end in other words (step 2610).

[0190]

Here, when there is a untreated collation result, it returns to the above-mentioned step 2602, and this step or subsequent ones is performed, on the other hand, when all are processed, even the last of a text file is written out to the file for an output as it is (step 2611).

[0191]

In this 5th Example, the attribute taken out by refer of an attribute can be processed by using an external general-purpose arithmetic means. By setting up the process result as attribute information from the attribute information designation section 2320, it can add in a document.

[0192]

According to 5th Example as explained above A pattern matching with the pattern using the relationship between the objects in a structuring document and the pattern of the hierarchical structure which constitutes a reference standard in other words is implemented by setting up pattern description information, and an attribute name, an attribute name and attribute value, since it was made to perform the refer or

要素)の属性の参照又は変更操作を実行するようにしたので、属性の参照又は変更の処理を自動化することができる。またこのとき、従来の如く予めスタイルを設定しておく必要がないので、ユーザの作業量を軽減させることができる。

【0193】

また特定部分の属性に、他の部分の属性値を元にした演算結果を設定するようなことも可能となる。

【0194】

【発明の効果】

以上説明したように、第1の発明によれば、解釈手段によって解釈された文書構造パターンと、再編成手段によって再編成された構造化文書とを照合手段により照合し、更に出力処理手段が、その照合により一致した文書構成要素を前記構造化文書から抽出し出力するようにしているので、構造化文書に対する指定された階層構造パターンに従った文書構成要素の検索処理を行うことができるという利点がある。

【0195】

第2の発明によれば、出力処理手段は、照合手段の照合により

alteration operation of the attribute of a part (document component) which matched, refer of an attribute or processing of alteration is automatable.

Moreover, since it is not necessary to set up a style beforehand in accordance with the prior art at this time, a user's workload can be lightened.

[0193]

Moreover, the calculation result made into the attribute of a specific part based on the attribute value of other parts can also be set up.

[0194]

[ADVANTAGE of the Invention]

As explained above, according to 1st invention, the document structure pattern interpreted by interpretation means and the structuring document reorganized by reorganization means are compared by collation means, furthermore, output-processing means extract the document component which was in agreement with the collation from said structuring document, and he is trying to output it.

Therefore, there is an advantage that the search of a document component according to the designated hierarchical-structure pattern with respect to a structuring document can be processed.

[0195]

According to 2nd invention, output-processing means, he is trying to output, after processing

一致した文書構成要素に対して、指定手段により指定された所定の処理例えば削除、置換、複写などの処理を施した後、出力するようにしているので、構造化文書に対する指定された階層構造パターンに従った文書構成要素に対して、削除、置換、複写などの処理を自動的に実行することができることとなり、構造化文書のバッチ処理が可能となる。

【0196】

第3の発明によれば、指定手段により削除処理が指定されると、出力処理手段は、蓄積手段に蓄積されている、照合手段の照合により一致した文書構成要素の親である文書構成要素から、蓄積手段に蓄積されている照合手段の照合により一致した文書構成要素に関する情報を取り除くと共に、当該親である文書構成要素からは削除されない子供の文書構成要素を出力するようにしているので、削除すべき文書構成要素の親の文書構成要素の内容を自動的に変更することができることとなり、構造化文書に対する編集操作を高率良く行うことができるという利点がある。

the fixed processing, for example, deletion, displacement, reproduction, etc., designated by designation means to the document component which was in agreement with collation of collation means.

Therefore, processing of deletion, displacement, reproduction, etc. can be automatically performed to the document component according to the designated hierarchical-structure pattern with respect to a structuring document, the batch processing of a structuring document can be performed.

[0196]

According to 3rd invention

If deletion processing is designated by designation means, while output-processing means will remove the information about the document component which was in agreement with collation of the collation means accumulated in accumulation means from the document component which is the parents of the document component which was in agreement with collation of collation means accumulated in accumulation means, a child's document component which is not deleted is outputted from the document component which is said parents.

Therefore, the content of the document component of the parents of a document component who should delete can be altered automatically, there is an advantage that the edit operation with respect to a structuring document can be performed with sufficient high rate.

【0197】

第4の発明によれば、解釈手段が、基準となる文書構成要素同志の接続関係を解釈し、また再編成手段が、格納手段に格納されているファイル内の複数の構造化文書それぞれを照合処理可能な形式の構造に再編成し、また照合手段が、解釈手段による解釈結果と、再編成手段による再編成結果とを照合し、更に出力処理手段が、照合手段の照合により一致した文書構成要素を前記再編成結果から抽出するようにしているので、複数の構造化文書から、指定された階層構造に従った文書構成要素を検索し出力することができるという利点がある。

【0198】

第5の発明によれば、複数の構造化文書を保存した複数のファイルを対象として、文書構成要素同志の接続関係に適合する文書構成要素を再編成結果から抽出するようにしているので、複数のファイルそれぞれに複数の構造化文書が存在している場合であっても、複数のファイルおよび複数の構造化文書に跨がって、指定された階層構造に従った文書構成要素を検索し出力することができることとなり、複

[0197]

According to 4th invention

Interpretation means interpret the relation of connection of the document component element who becomes a reference standard, moreover, reorganization means reorganize two or more structuring documents of each in the file stored in storing means in the structure of the form which can carry out collation processing, moreover, collation means compare the interpretation result by interpretation means, and the reorganization result by reorganization means, furthermore, output-processing means extract the document component which was in agreement with collation of collation means from said reorganization result.

Therefore, there is an advantage that the document component according to the designated hierarchical structure can be searched and outputted from two or more structuring documents.

[0198]

According to 5th invention, the document component which adapts a document component element's relation of connection is extracted from a reorganization result by making into object two or more files which conserved two or more structuring documents.

Therefore, even if it is the case where two or more structuring documents exist in two or more files of each, ranging over two or more file and two or more structuring documents, the document component according to the designated hierarchical structure can be searched and outputted, there is an advantage

数の文書に対する検索処理を高速に実行することができるという利点がある。

【0199】

第6の発明によれば、ソース側及びターゲット側それぞれの複数の構造化文書を保存した複数のファイルを対象として、文書構成要素同志の接続関係に適合する文書構成要素を再編成結果から抽出すると共に、ターゲット側の抽出結果である文書構成要素に対するソース側の抽出結果である文書構成要素の挿入を実行するようにしているので、ターゲット側の複数のファイルそれぞれに保存されている複数の構造化文書から抽出された文書構成要素に対する、ソース側の複数のファイルそれぞれに保存されている複数の構造化文書から抽出された文書構成要素の挿入を一度に行うことができる。よって文書の編集処理を迅速に行うことができるという利点がある。

【0200】

第7の発明によれば、出力処理手段は、指定された属性に関する情報に基づいて、照合手段の照合により一致した文書構成要素の属性の参照又は変更の操作を実行するようにしているので、構造化文書から、指定され

that search processing with respect to two or more documents can be performed at high speed.

[0199]

According to 6th invention, by making into object two or more files as for which the source and target side conserved each structuring documents of two or more, while extracting the document component which adapts a document component element's relation of connection from a reorganization result, it is made to perform insertion of the document component which is an extraction result by the side of the source with respect to the document component which is an extraction result by the side of a target.

Therefore, the document component extracted from two or more structuring documents with respect to the document component extracted from two or more structuring documents conserved at two or more files of each by the side of a target conserved at two or more files of each by the side of a source can be inserted at once.

Therefore, there is an advantage that edit of a document can be processed rapidly.

[0200]

According to 7th invention, output-processing means are made to perform refer of the attribute of the document component which was in agreement with collation of collation means, or operation of alteration based on the information about the designated attribute.

Therefore, the document component which

た階層構造に従った文書構成要素を検索し、この文書構成要素の属性に対する参照又は変更の操作を実施することができることとなり、構造化文書の文書構成要素の属性に対する操作を容易に実施することができるという利点がある。

followed the designated hierarchical structure from the structuring document is searched, the operation of refer or alteration with respect to the attribute of this document component can be implemented, there is an advantage that the operation with respect to the attribute of the document component of a structuring document can be implemented easily.

【図面の簡単な説明】**[BRIEF DESCRIPTION OF THE DRAWINGS]****【図 1】**

本発明に係る文書処理装置の第 1 の実施例を示す機能ブロック図。

[FIG 1]

The functional-block figure which shows 1st Example of the word-processing apparatus based on this invention.

【図 2】

図 1 に示した実施例の装置を実現するためのハードウェア構成を示すブロック図。

[FIG 2]

The block diagram which shows the hardware composition for implementing the apparatus of the Example shown in FIG. 1.

【図 3】

第 1 の実施例における解釈部によるパターン記述の解釈処理を説明するための図。

[FIG 3]

The figure for demonstrating interpretation processing of the pattern description by the interpretation section in 1st Example.

【図 4】

第 1 の実施例における解釈部の解釈処理動作を示すフローチャート。

[FIG 4]

The flowchart which shows the interpretation processing operation of the interpretation section in 1st Example.

【図 5】

第 1 の実施例における解釈部によるパターン記述の解釈処理過程を説明するための図。

[FIG 5]

The figure for demonstrating the interpretation process stage of the pattern description by the interpretation section in 1st Example.

【図 6】

第 1 の実施例における解釈部によるパターン記述の解釈処理過程を説明するための図。

[FIG 6]

The figure for demonstrating the interpretation process stage of the pattern description by the interpretation section in 1st Example.

【図 7】

第 1 の実施例における再編成部による構造化文書の再編成処理を説明するための図。

[FIG 7]

The figure for demonstrating reorganization processing of the structuring document by the reorganization section in 1st Example.

【図 8】

第 1 の実施例における照合部の照合処理動作を示すフローチャート。

[FIG 8]

The flowchart which shows the collation processing operation of the collation section in 1st Example.

【図 9】

第 1 の実施例における照合部による文書構造パターンと再編成構造化文書との照合処理を説明するための図。

[FIG 9]

The figure for demonstrating collation processing with the document structure pattern and reorganization structuring document by the collation section in 1st Example.

【図 10】

本発明に係る文書処理装置の第 2 の実施例を示す機能ブロック図。

[FIG 10]

The functional-block figure which shows 2nd Example of the word-processing apparatus based on this invention.

【図 11】

第 2 の実施例における解釈部によるパターン記述の解釈処理を説明するための図。

[FIG 11]

The figure for demonstrating interpretation processing of the pattern description by the interpretation section in 2nd Example.

【図 12】

第 2 の実施例における再編成部による構造化文書の再編成処理を説明するための図。

[FIG 12]

The figure for demonstrating reorganization processing of the structuring document by the reorganization section in 2nd Example.

【図 1 3】

第2の実施例における照合部による文書構造パターンと再編成構造化文書との照合処理を説明するための図。

[FIG 13]

The figure for demonstrating collation processing with the document structure pattern and reorganization structuring document by the collation section in 2nd Example.

【図 1 4】

第2の実施例における命令処理部の解釈処理を説明するための図。

[FIG 14]

The figure for demonstrating interpretation processing of the command-processing section in 2nd Example.

【図 1 5】

第2の実施例における出力処理部の出力処理を説明するための図。

[FIG 15]

The figure for demonstrating output processing of the output-processing section in 2nd Example.

【図 1 6】

本発明に係る文書処理装置の第3の実施例を示す機能ブロック図。

[FIG 16]

The functional-block figure which shows 3rd Example of the word-processing apparatus based on this invention.

【図 1 7】

第3実施例の削除情報の構造の一例を示す図。

[FIG 17]

The figure which shows an example of the structure of the deletion information on 3rd Example.

【図 1 8】

第3実施例における削除情報作成処理を示すフローチャート。

[FIG 18]

The flowchart which shows the deletion information creation processing in 3rd Example.

【図 1 9】

第3実施例における出力処理部による出力処理動作を示すフローチャート。

[FIG 19]

The flowchart which shows the output processing operation by the output-processing section in 3rd Example.

【図 2 0】

第3実施例における出力処理部による出力処理動作を示すサブルーチン。

[FIG. 20]

The subroutine which shows the output processing operation by the output-processing section in 3rd Example.

【図 2 1】

本発明に係る文書処理装置の第4の実施例を示す機能ブロック図。

[FIG. 21]

The functional-block figure which shows 4th Example of the word-processing apparatus based on this invention.

【図 2 2】

第4実施例における構造化文書を説明するための図。

[FIG. 22]

The figure for demonstrating the structuring document in 4th Example.

【図 2 3】

第4実施例における抽出点認識処理動作を示すフローチャート。

[FIG. 23]

The flowchart which shows the extracting point-of-view recognition processing operation in 4th Example.

【図 2 4】

第4の実施例における照合部による文書構造パターンと再編成構造化文書との照合処理を説明するための図。

[FIG. 24]

The figure for demonstrating collation processing with the document structure pattern and reorganization structuring document by the collation section in 4th Example.

【図 2 5】

第4の実施例における出力処理部の出力処理動作を示すフローチャート。

[FIG. 25]

The flowchart which shows the output processing operation of the output-processing section in 4th Example.

【図 2 6】

第4の実施例における出力処理部の出力処理を説明するための図。

[FIG. 26]

The figure for demonstrating output processing of the output-processing section in 4th Example.

【図 2 7】**[FIG. 27]**

第4の実施例における出力処理部の出力処理を説明するための図。

The figure for demonstrating output processing of the output-processing section in 4th Example.

【図28】

本発明に係る文書処理装置の第5の実施例を示す機能ブロック図。

[FIG 28]

The functional-block figure which shows 5th Example of the word-processing apparatus based on this invention.

【図29】

第5実施例における構造化文書を説明するための図。

[FIG 29]

The figure for demonstrating the structuring document in 5th Example.

【図30】

第5実施例における文書構成要素の属性の参照処理動作を示すフローチャート。

[FIG 30]

The flowchart which shows the refer processing operation of the attribute of the document component in 5th Example.

【図31】

第5実施例における文書構成要素の属性の変更処理動作を示すフローチャート。

[FIG 31]

The flowchart which shows the alteration processing operation of the attribute of the document component in 5th Example.

【符号の説明】

1 0、1 6 1 0…メモリ、2 0、1 6 1 1…パターン記述情報、3 0、1 6 2 0…解釈部、4 0、1 6 1 2…文書構造パターン、5 0、1 6 3 0…再編成部、6 0…文書ファイル、7 0…再編成構造化文書、8 0、1 6 6 0…照合部、9 0、1 0 3 0、1 6 9 0、2 3 3 0…出力処理部、2 1 0…中央処理装置、2 2 0…主メモリ、2 3 0…ディスク、2 4 0…ディスプレイ、2 5 0

[Description of Symbols]

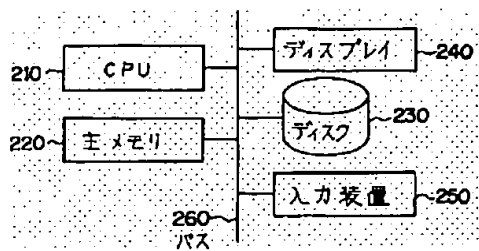
10, 1610... Memory, 20, 1611... Pattern description information, 30, 1620... Interpretation section, 40, 1612... Document structure pattern, 50, 1630... Reorganization section, 60... Text file 70... Reorganization structuring document, 80, 1660... Collation section, 90, 1030, 1690, 2330... Output-processing section, 210... Central processing unit, 220... Main memory 230... Disc, 240... Display, 250... Input device, 1010... Compatible node storage part, 1020... Command-processing section, 1040... Deletion

…入力装置、1 0 1 0…適合ノード蓄積部、1 0 2 0…命令処理部、1 0 4 0…削除情報蓄積部、1 6 1 3…ソース再編成構造化文書群、1 6 1 4…ターゲット再編成構造化文書群、1 6 4 0…ソース文書ファイル群、1 6 5 0…ターゲット文書ファイル群、1 6 6 1…複数ファイル抽出点認識部、1 6 6 2…複数ファイル挿入点認識部、1 6 7 0…複数抽出点蓄積部、1 6 8 0…複数挿入点蓄積部、1 6 9 1…複数ファイル部分構造抽出挿入部、1 7 0 0…抽出挿入関係ルール指定部、2 3 1 0…ファイル位置情報保持部、2 3 2 0…属性情報指定部。

information storage part, 1613... Source reorganization structuring document group, 1614... Target reorganization structuring document group, 1640... Source text-file group, 1650... Target text-file group, 1661... two or more files extracting point recognition part, 1662... two or more files inserting point recognition part, 1670... two or more extracting point storage part, 1680... two or more inserting point storage part, 1691... Two or more files partial structure extraction insertion part, 1700... Extraction insertion relationship rule designation section, 2310... File positional-information retainer, 2320... Attribute information designation section.

【図 2】

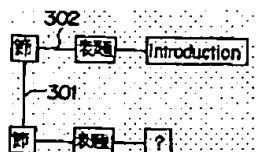
[FIG 2]



220 Main memory
 240 Display
 230 Disc
 250 Input device
 260 Bus

【図 3】

[FIG. 3]

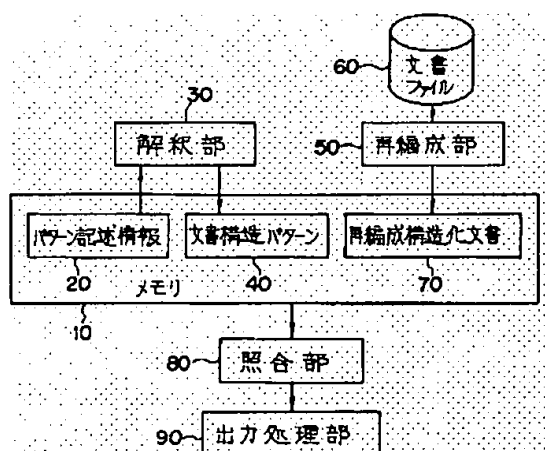


302 Clause Title

301 Clause Title

【図 1】

[FIG. 1]



60 Text file

30 Interpretation section

50 Reorganization section

20 Pattern description information

40 Document structure pattern

70 Reorganization structuring document

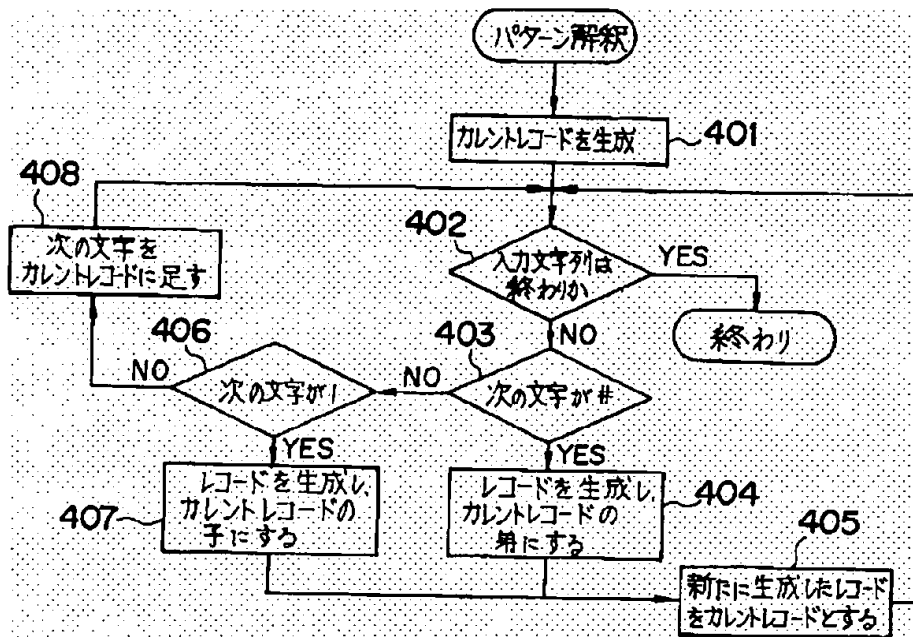
10 Memory

80 Collation section

90 Output-processing section

【図 4】

[FIG. 4]



Pattern interpretation

401: Current coding was generated.

408: Adding the next character to current coding.

406: The next character is 1.

402: Is an input string end? End

403: The next character is #.

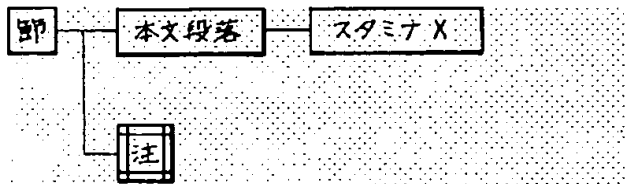
407: Generating a record and making it the child of a current record.

404: Generating a record and making it the younger brother of current coding.

405: Let the newly generated record be a current record.

【図 11】

[FIG. 11]



Clause

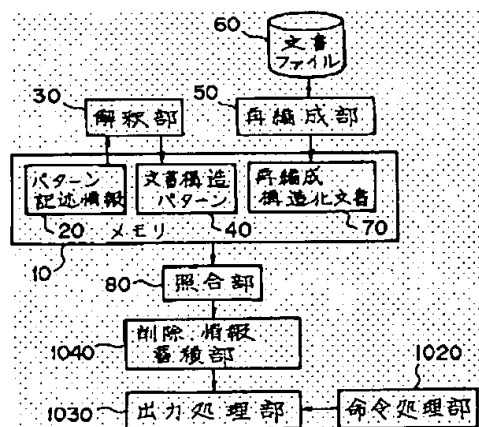
This stage

Stamina X

Notes

【図 16】

[FIG 16]



60: Text file

30: Interpretation section

50: Reorganization section

20: Pattern description information

40: Document structure pattern

70: Reorganization structuring document

10: Memory

80: Collation section

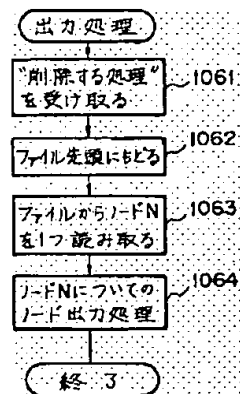
1040: Deletion, an information storage part

1030: Output-processing section

1020: Command-processing section

【図 19】

[FIG. 19]



Output processing

1061: "Receiving the processing to delete."

1062: Returning to a file head.

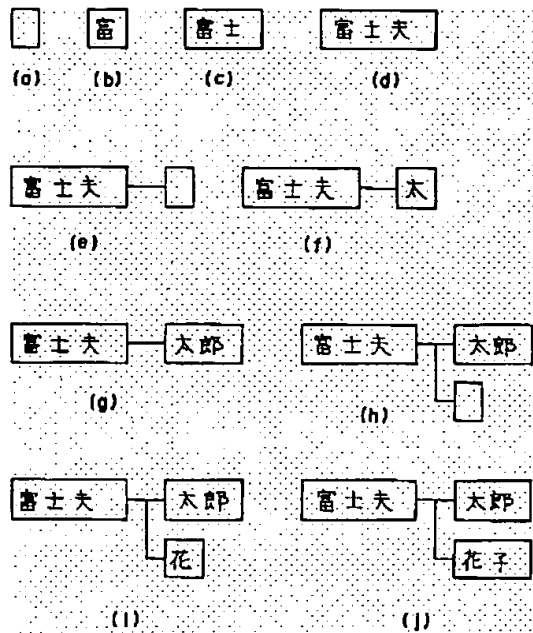
1063: Reading one node N in a file.

1064: Node output processing about Node N

Completion

【図 5】

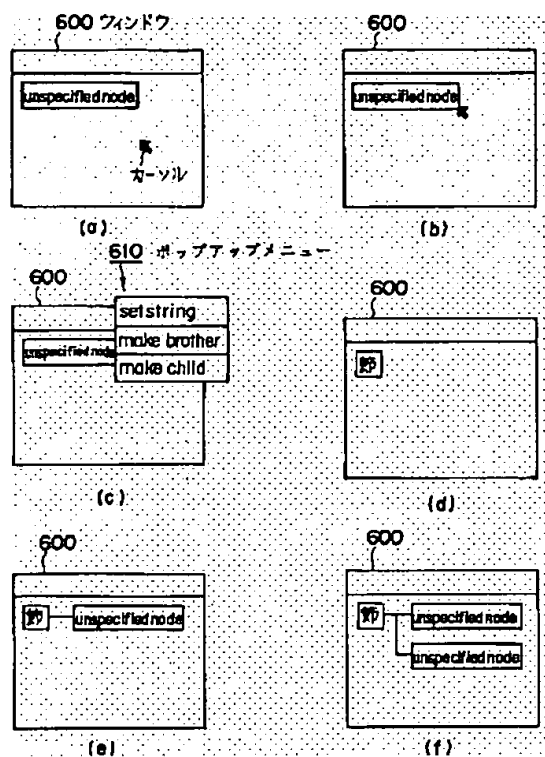
[FIG. 5]



- (b): Fu
 (c): Fuji
 (d): Fujio
 (e): Fujio
 (f): Fujio - Ta
 (g): Fujio-Taro
 (h): Fujio-Taro
 (i): Fujio-Taro-Hana
 (j): Fujio-Taro-Hanako

【図 6】

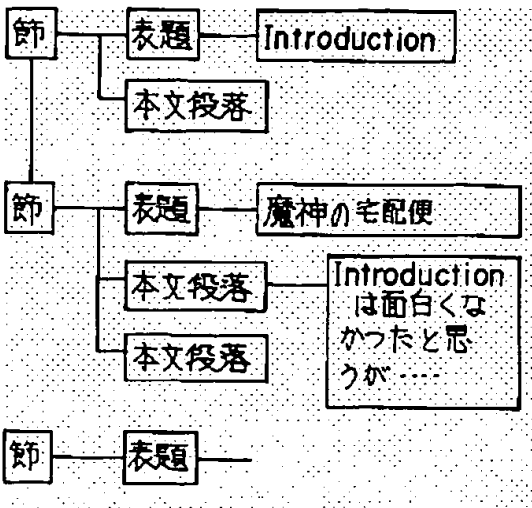
[FIG. 6]



- (a): 600 windows
Cursor
- (c): 610 popup menus
- (d), (e), (f): Clause

【図 7】

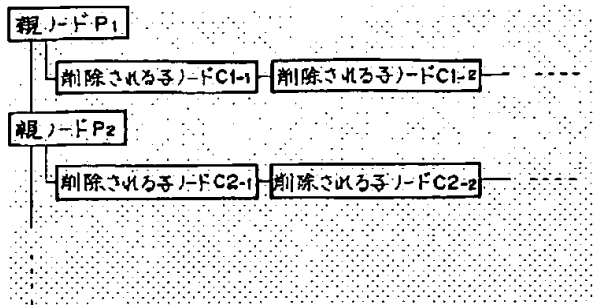
[FIG. 7]



Clause Title
This paragraph
Clause Title Parcel delivery service of fiend
This paragraph Although it thinks that Introduction was not interesting
This paragraph
Clause Title


【図 17】

[FIG. 17]

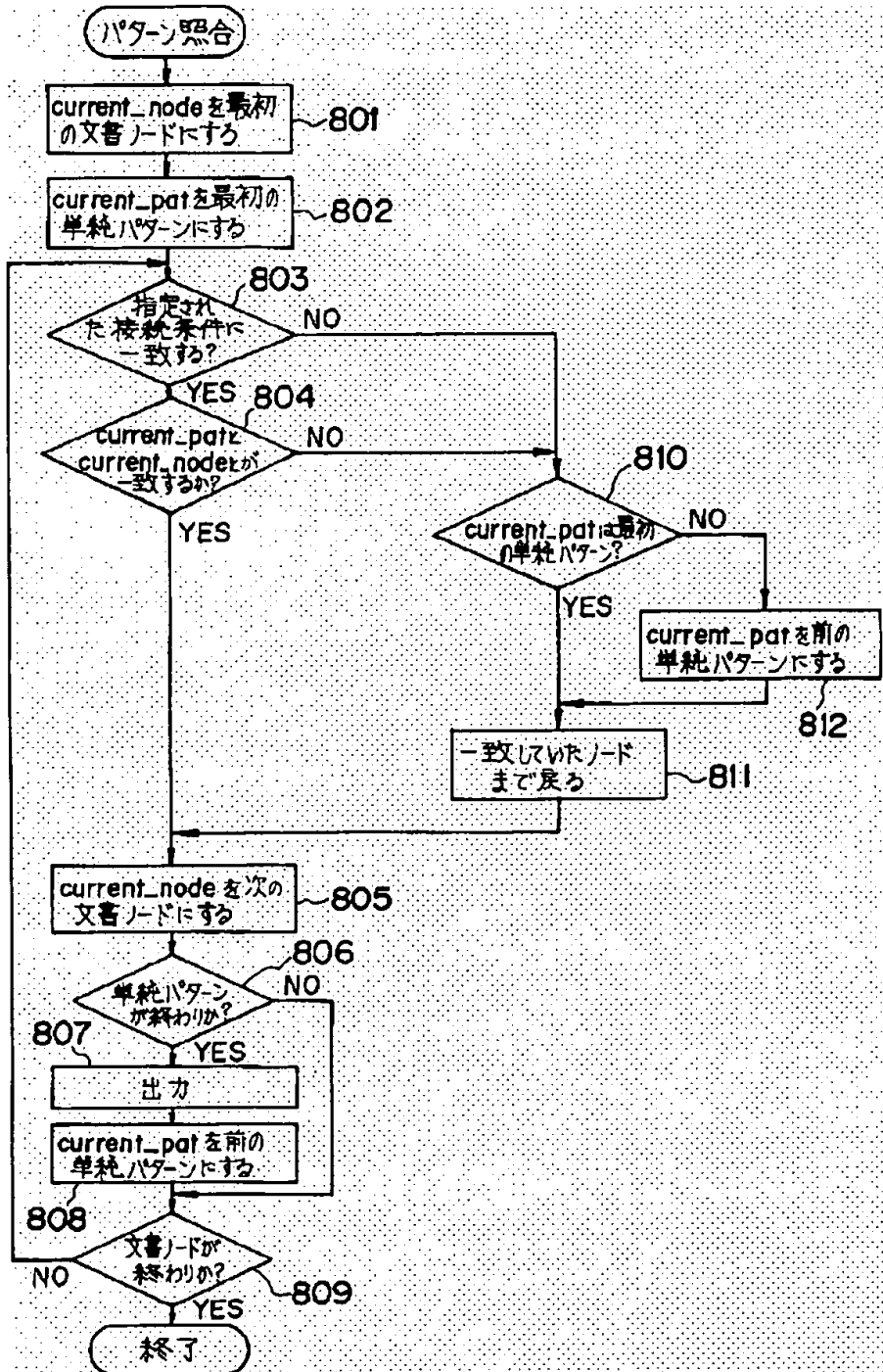


Parent node P1

The child node deleted C1-1 Child node deleted C1-2
Parent node P2
The child node deleted C2-1 Child node deleted C2-2

【 8】

[FIG. 8]



Pattern match

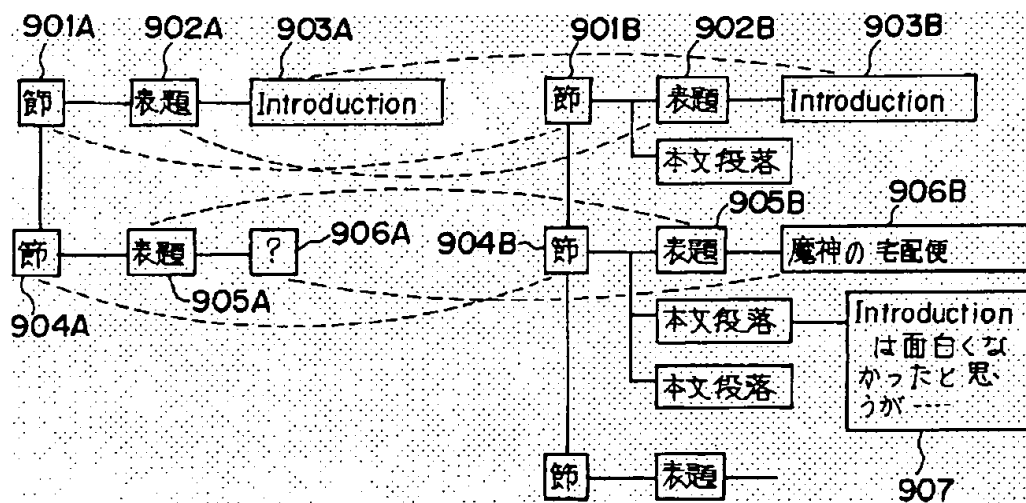
801: Making current_node into the first document node.

802: Using current_pat as the first simple pattern.

803: Is it in agreement with the designated connection conditions?
 804: Are current_pat and current_node in agreement?
 805: Making current_node into the next document node.
 806: Is finally a simple pattern?
 807: Output
 808: Using current_pat as a previous simple pattern.
 809: Is a document node end?
 Completion
 810: current_pat is the first simple pattern?
 812: Using current_pat as a previous simple pattern.
 811: Returning to the congruous nodes.

【図 9】

[FIG 9]



901A, 904A, 901B, 904B,: Clause

902A, 905A, 902B, 905B: Title

906B: Parcel delivery service of fiend

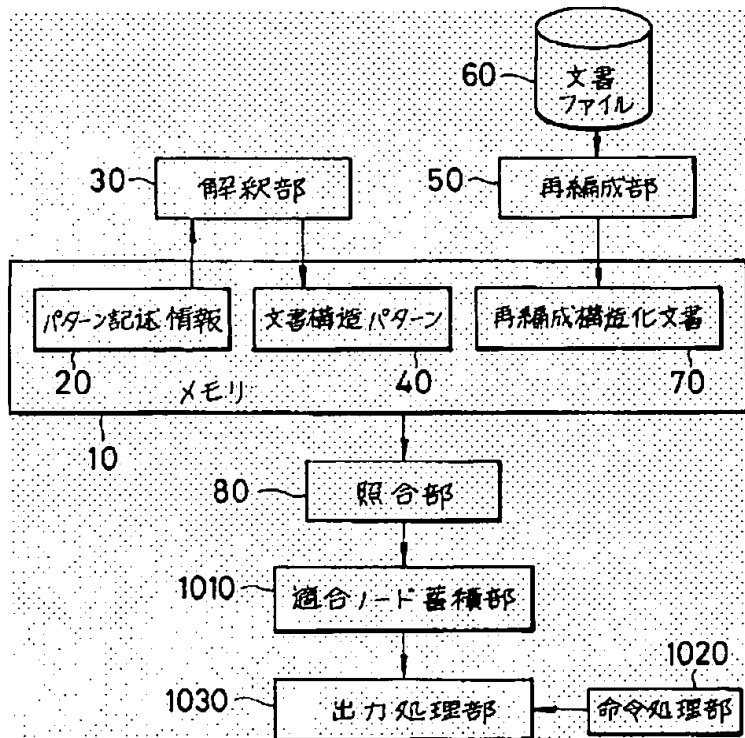
907: Although it thinks that Introduction was not interesting ...

This paragraph

Title

【図10】

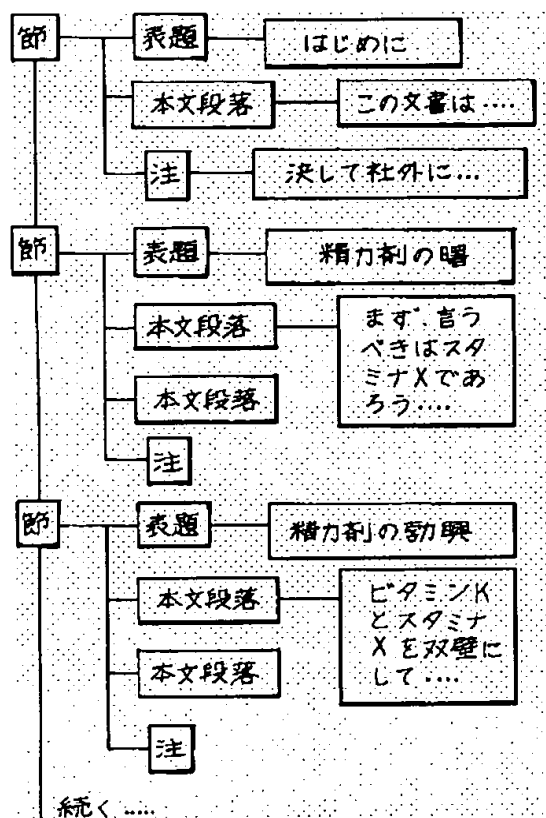
[FIG 10]



- 60 Text file
- 30 Interpretation section
- 50 Reorganization section
- 20 Pattern description information
- 40 Document structure pattern
- 70 Reorganization structuring document
- 10 Memory
- 80 Collation section
- 1010: Compatible node storage part
- 1030: Output-processing section
- 1020: Command-processing section

【図 12】

[FIG 12]



Clause Title Introduction

This paragraph This document ...

Notes It is never external...

Clause Title Dawn of an energy agent

This paragraph First, the power to say is stamina...

This paragraph

Notes

Clause Title The sudden rise of an energy agent

This paragraph Vitamin K and stamina X are made into two greatest authorities...

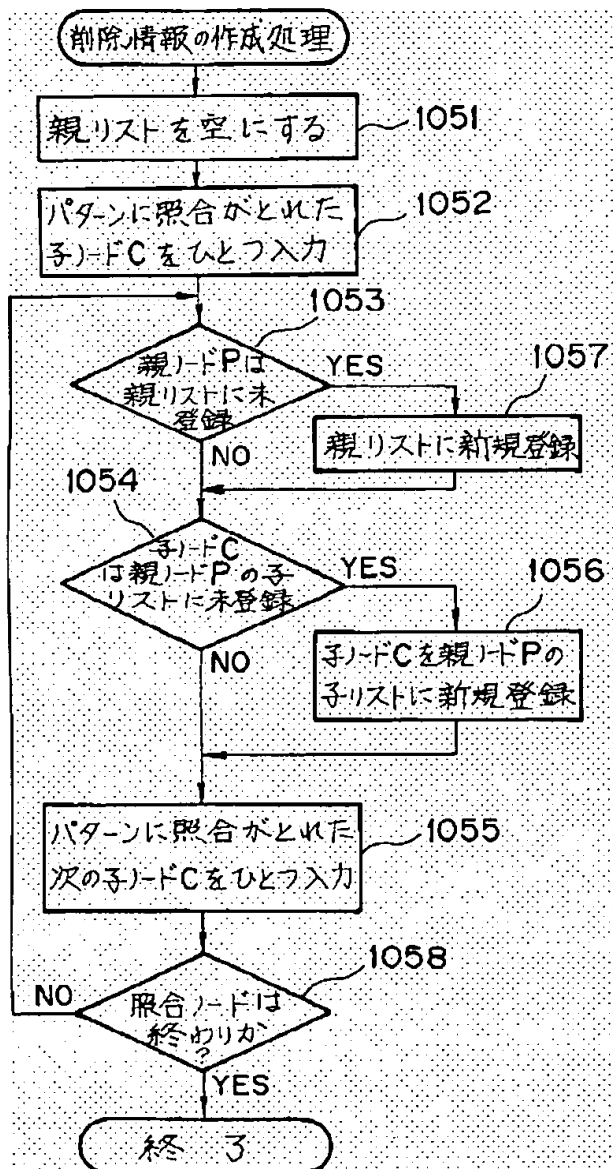
This paragraph

Notes

It continues...

【図18】

[FIG. 18]



Creation processing of deletion information

1051: Emptying a parent list.

1052: Child node C which was able to take collation was one-input into the pattern.

1053: Parent node P was un-registered into the parent list.

1057: new registration to a parent list.

1054: Child node C was un-registered into the child list of parent node P.

1056: It is new registration to the child list of parent node P about child node C.

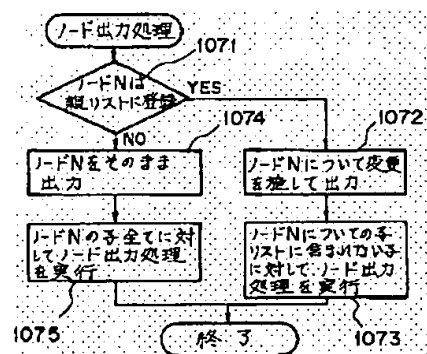
1055: The next child node C which was able to take the pattern match was one-input.

1058: Is a collation node end?

Completion

【図 20】

[FIG. 20]



Node output processing

1071: Node N was registered into the parent list.

1072: Altering about Node N and it is an output.

1073: Node output processing was performed to the child who is not contained in the child list about Node N.

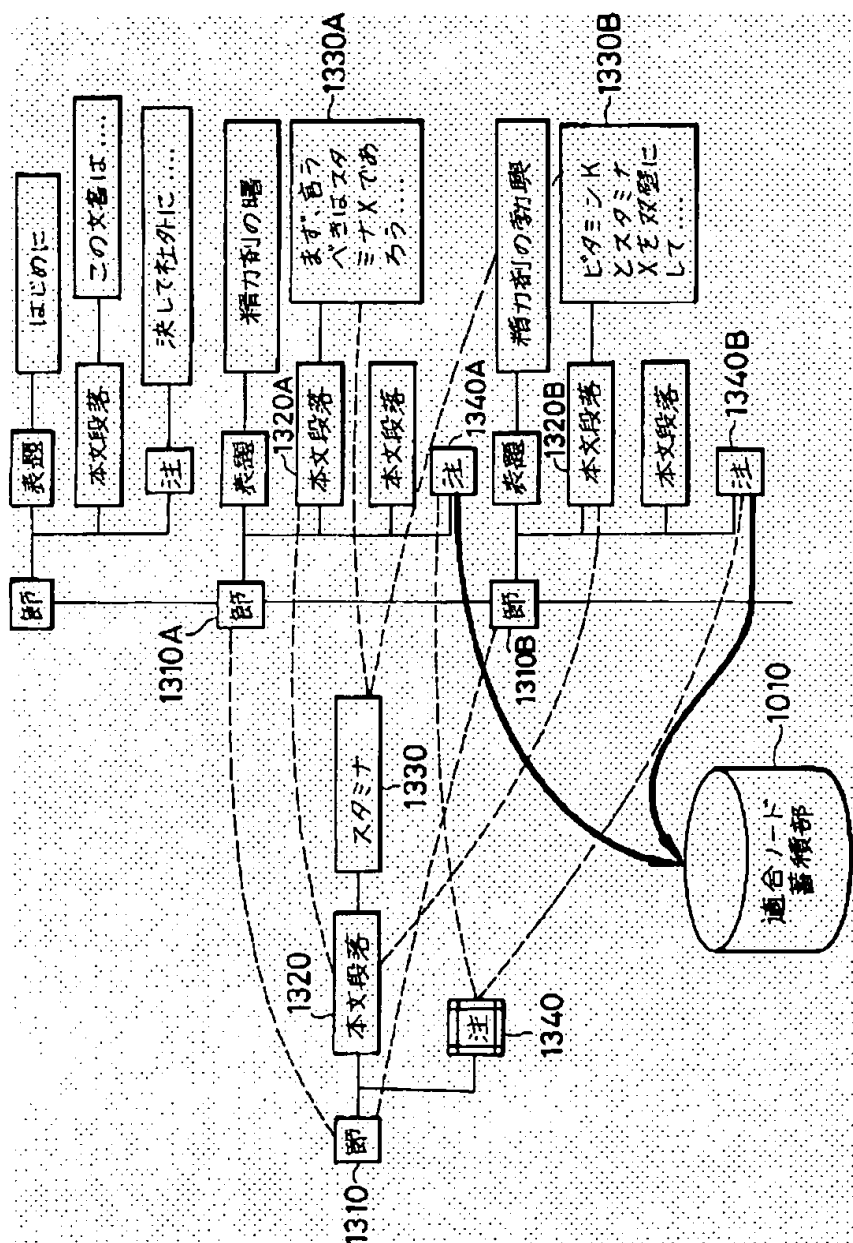
1074: Node N was outputted as it was.

1075: Node output processing was performed to all the children of Node N.

Completion

【図 13】

[FIG. 13]



1310, 1310A, 1310B: Clause

1320, 1320A, 1320B: This stage

1340, 1340A, 1340B: Notes

1330: Stamina

1330A: The power to say is stamina first...

1330B: Making a vitamin K and stamina X into two greatest authorities...

1010: Compatible node storage part

(Right-part)

Clause

Title

Introduction

This paragraph

This text ...

Notes

It is never external...

Title


Dawn of an energy agent

This paragraph

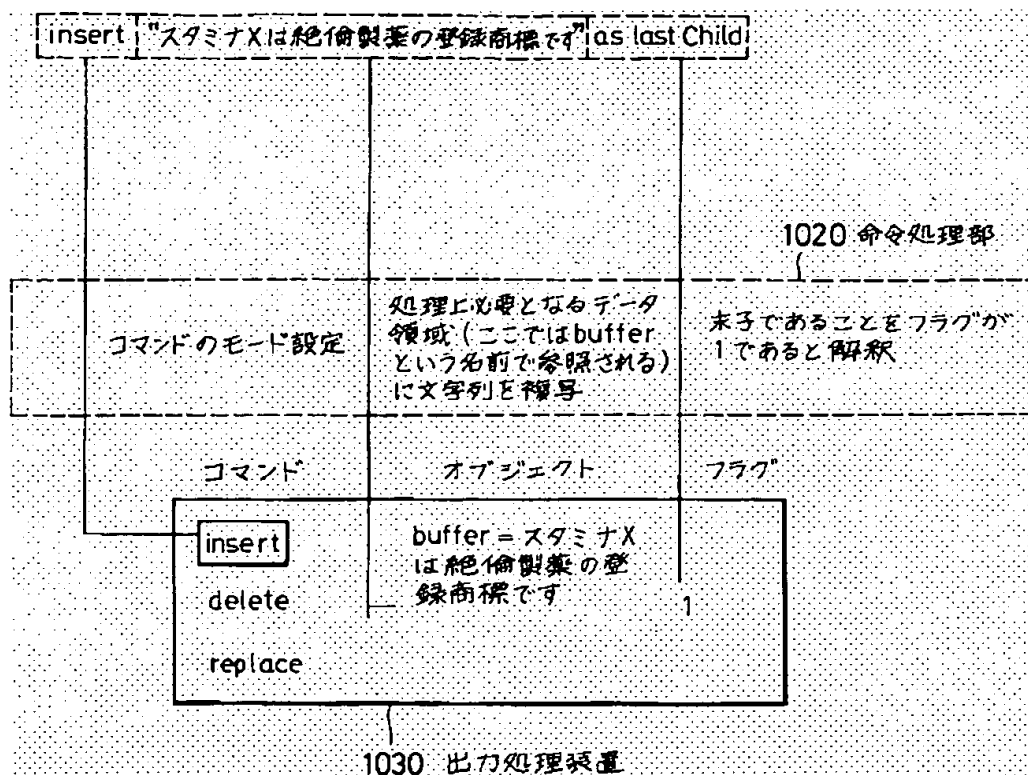
Title

The sudden rise of an energy agent

This paragraph

【 1 4】

[FIG. 14]



"Stamina X is the trademark of Zetsurin pharmaceutical."

A mode setup of a command

The character string was reproduced to the data area (here, seen by the name of buffer) which is needed on processing.

1020: Command-processing section

When the flag was 1, it interpreted that she was last-born child.

Command

Object

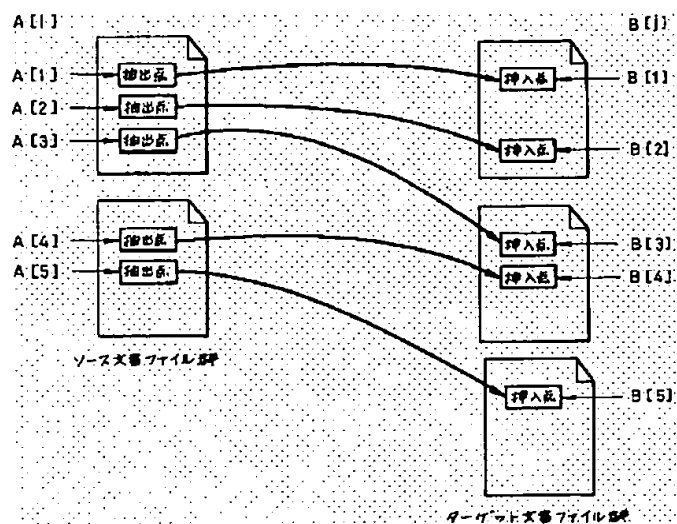
Flag

The buffer= stamina X is the trademark of Zetsurin pharmaceutical.

1030: Output processing apparatus

【図 26】

[FIG. 26]

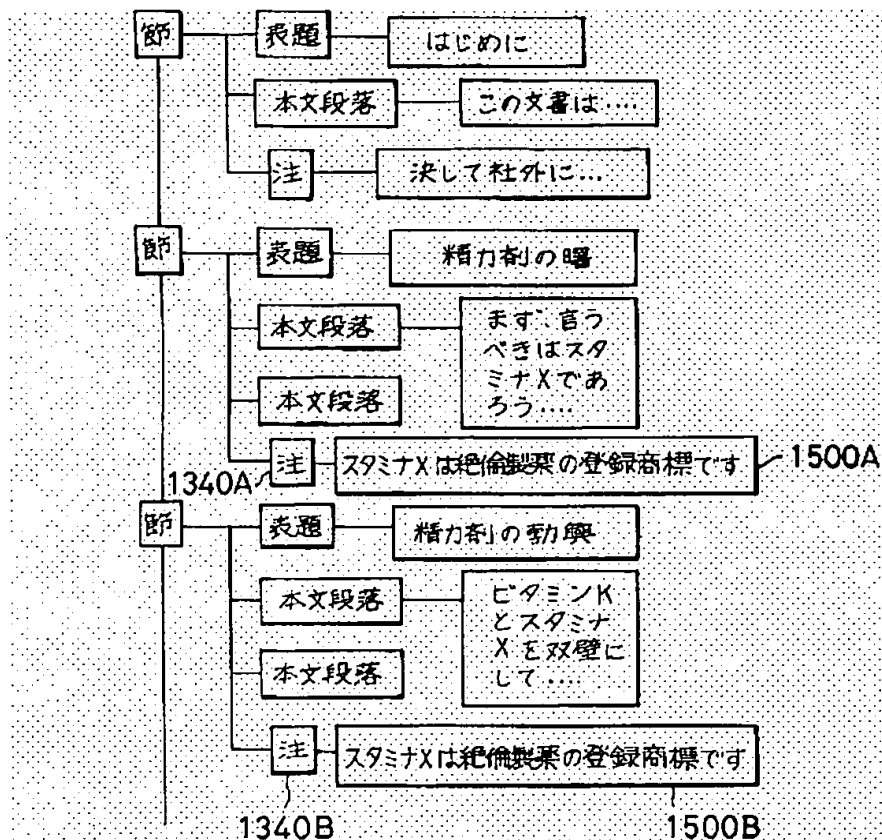


A: Extracting point
 Source text-file group

B: Insertion store
 Target text-file group

【図 15】

[FIG. 15]



Clause Title Introduction

This paragraph This document ...

Notes It is never external...

Clause Title Dawn of an energy agent

This paragraph First, the power to say is stamina...

This paragraph

1340A, Notes Stamina X is the trademark of Zetsurin pharmaceutical

Clause Title The sudden rise of an energy agent

This paragraph Vitamin K and stamina X are made into two greatest authorities...

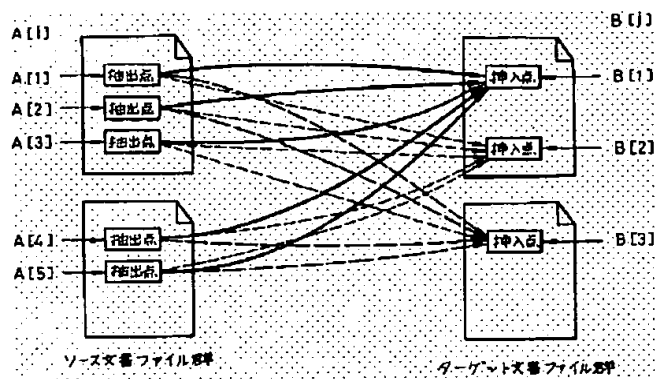
This paragraph

1340B, Notes Stamina X is the trademark of Zetsurin pharmaceutical

It continues...

【図 27】

[FIG. 27]

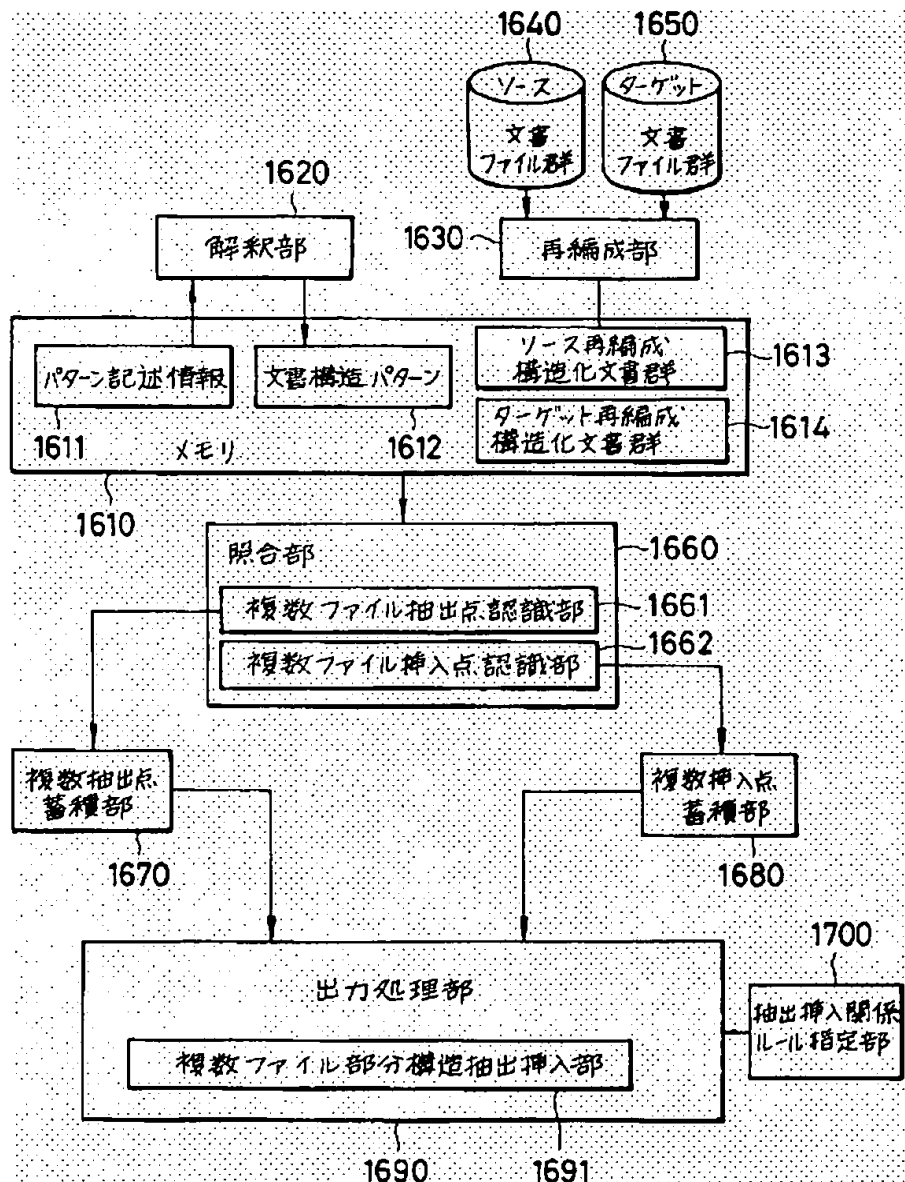


A: Extracting point
 Source text-file group

B: Insertion store
 Target text-file group

【図 21】

[FIG. 21]



1640: Source Text-file group

1650: Target Text-file group

1620: Interpretation section

1630: Reorganization section

1611: Pattern description information

Memory

1612: Document structure pattern

1613: Source reorganization structuring document group

1614: Target reorganization structuring document group

Collation section

1661: Two or more files extracting point recognition part

1662: Two or more files inserting point recognition part

1670: Two or more extracting point storage part

1680: Two or more inserting point storage part

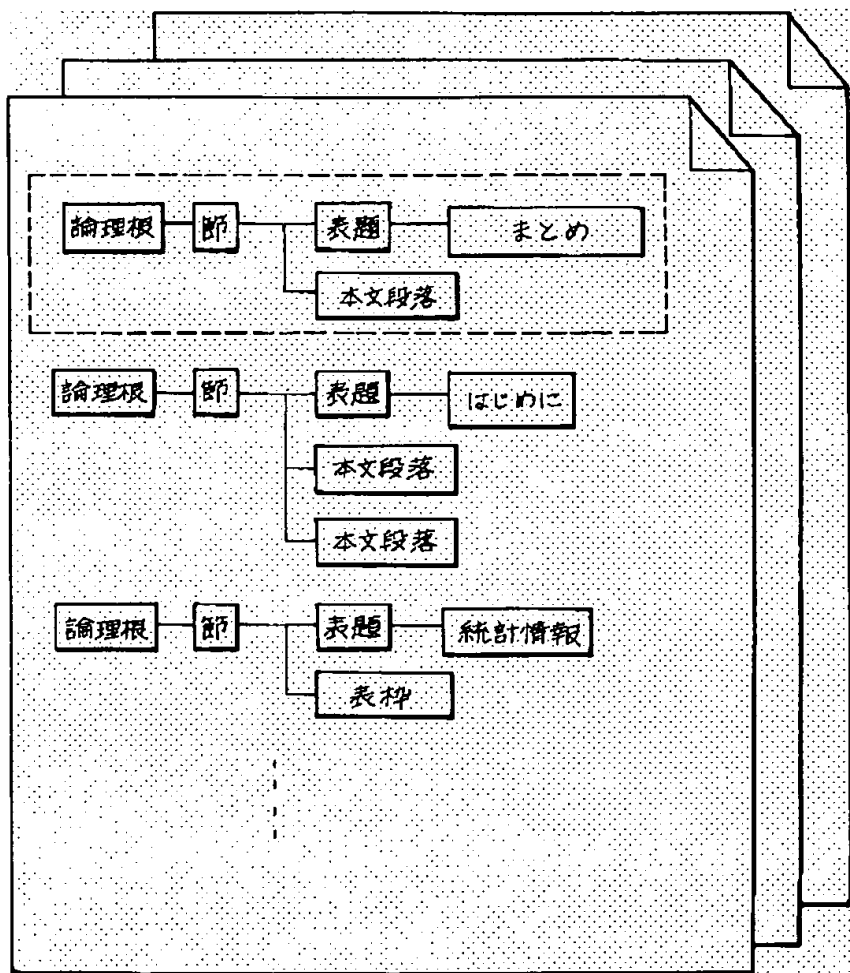
Output-processing section

1691: Two or more files partial structure extraction insertion part

1700: Extraction insertion relationship rule designation section

【図 22】

[FIG. 22]



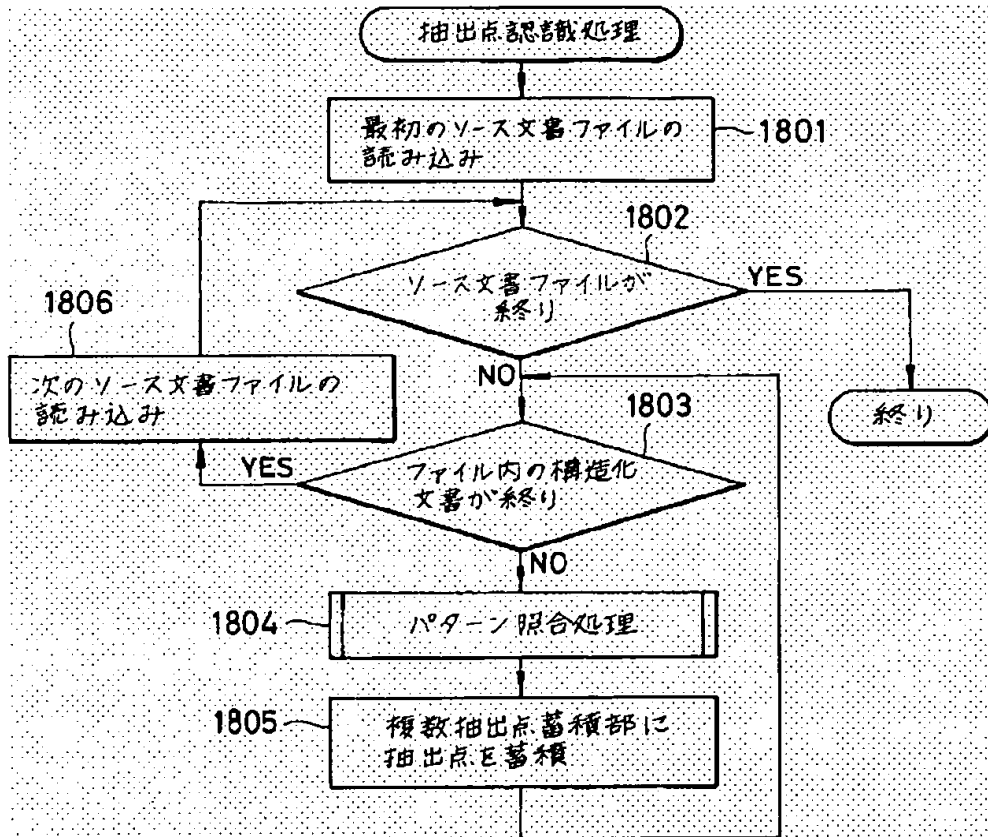
Logic solution Clause Title Conclusion
This paragraph

Logic solution Clause Title Introduction
This paragraph
This paragraph

Logic solution Clause Title Front frame
Statistic information

【図 23】

[FIG 23]



Extracting point recognition processing

1801: Reading of the first source text file

1802: A source text file finishes.

1806: Reading of the next source text file

1803: The structuring document in a file finishes.

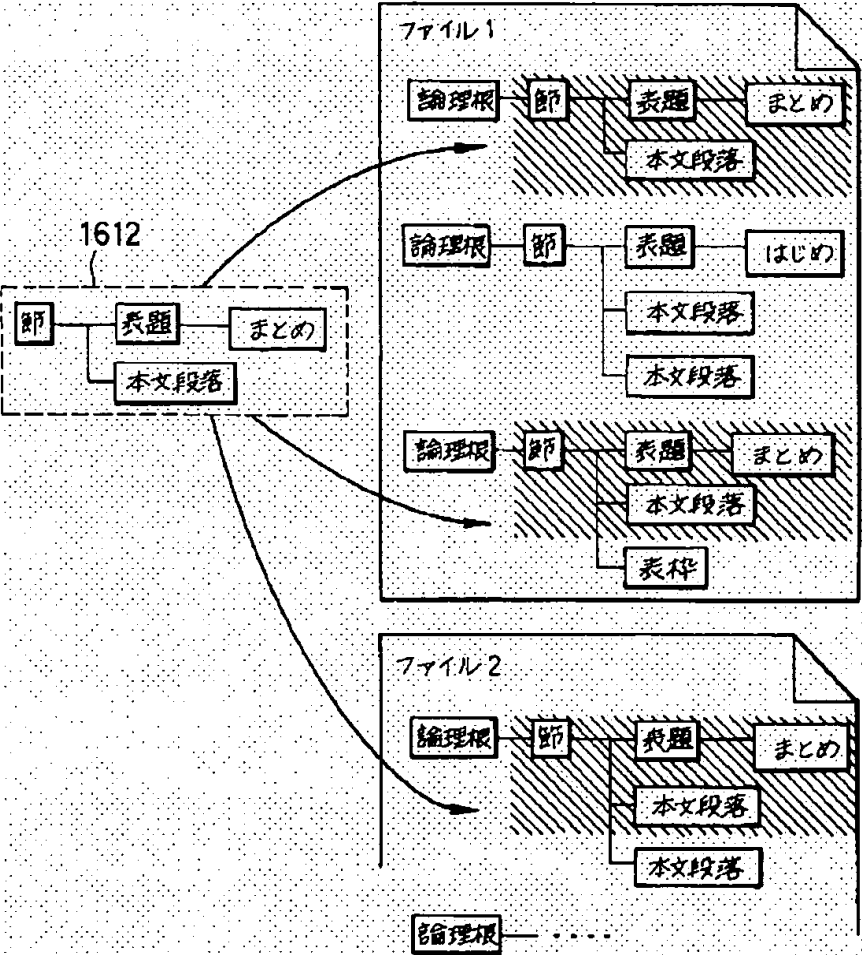
1804: Pattern-match processing

1805: The extracting point was accumulated in the two or more extracting point storage part.

End

【図 24】

[FIG 24]



1612: Clause Title Conclusion
This paragraph

File 1: Logic solution Clause Title Conclusion
This paragraph


Logic solution Clause Title Start
This paragraph
This paragraph

Logic solution Clause Title Conclusion

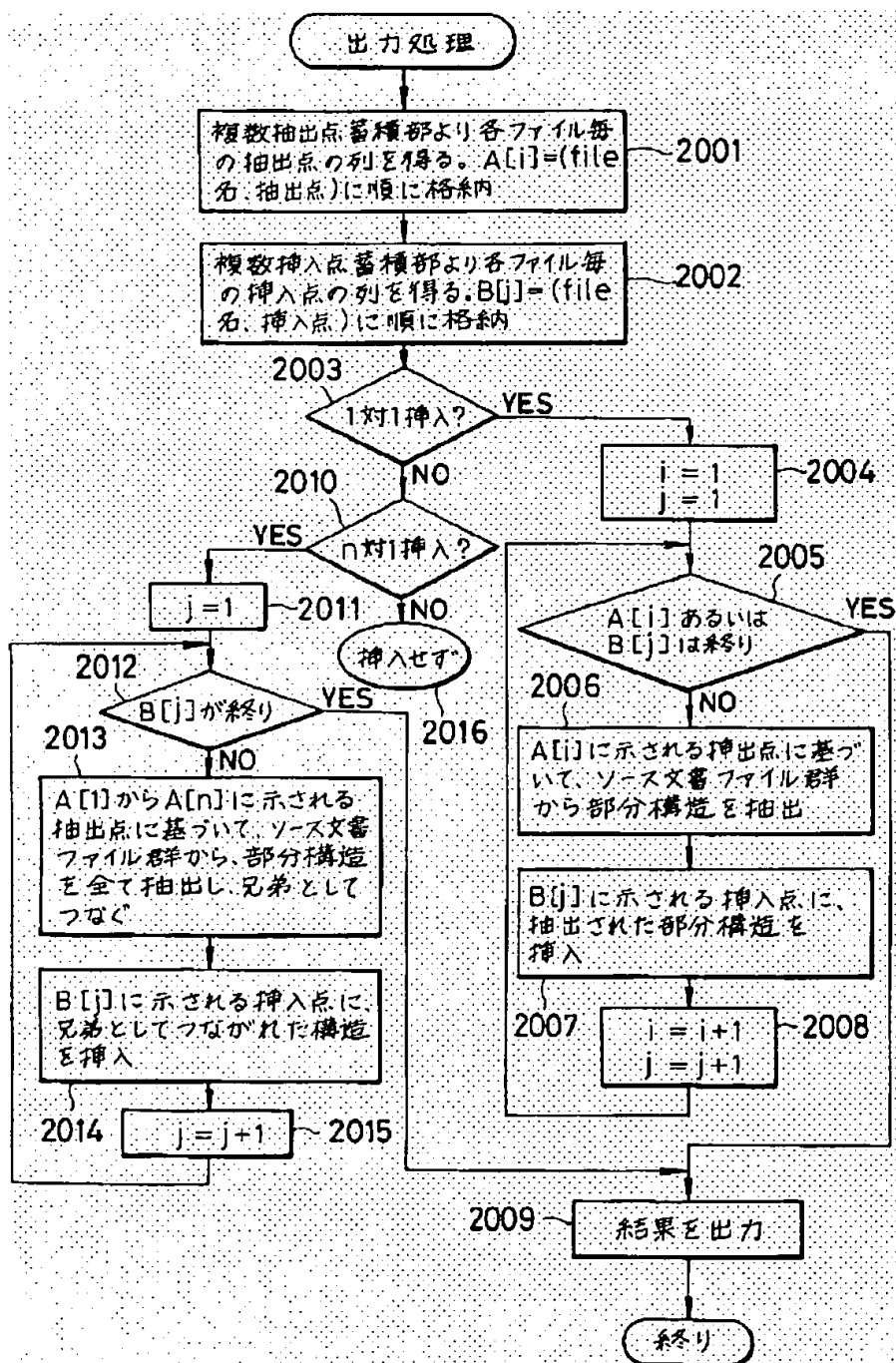
This paragraph
Front frame

File 2: Logic solution	Clause	Title	Start
	This paragraph		
	This paragraph		

Logic solution

【 2 5】

[FIG. 25]



Output processing

2001: Obtaining the row of the extracting point for every file from two or more extracting point storage part.

A[i]=

It stores in (a file name and an inserting point) in order.

2002: Obtaining the row of the extracting point for every file from two or more extracting point storage part.

B[j]=It stores in (a file name and an inserting point) in order.

2003: 1 to 1 insertion?

2010: n to 1 insertion?

2012: B [j] finishes.

2013: Based on the extracting point shown from A [1] to A [n], from a source text-file group, extracting all partial structure and tying as a brother.

2014: Inserting the structure connected with the inserting point shown by B [j] as a brother.

2016: Do not inserting.

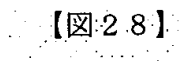
2005: A [j] or B [j] finishes.

2006: Based on the extracting point shown by A [j], the partial structure was extracted from the source text-file group.

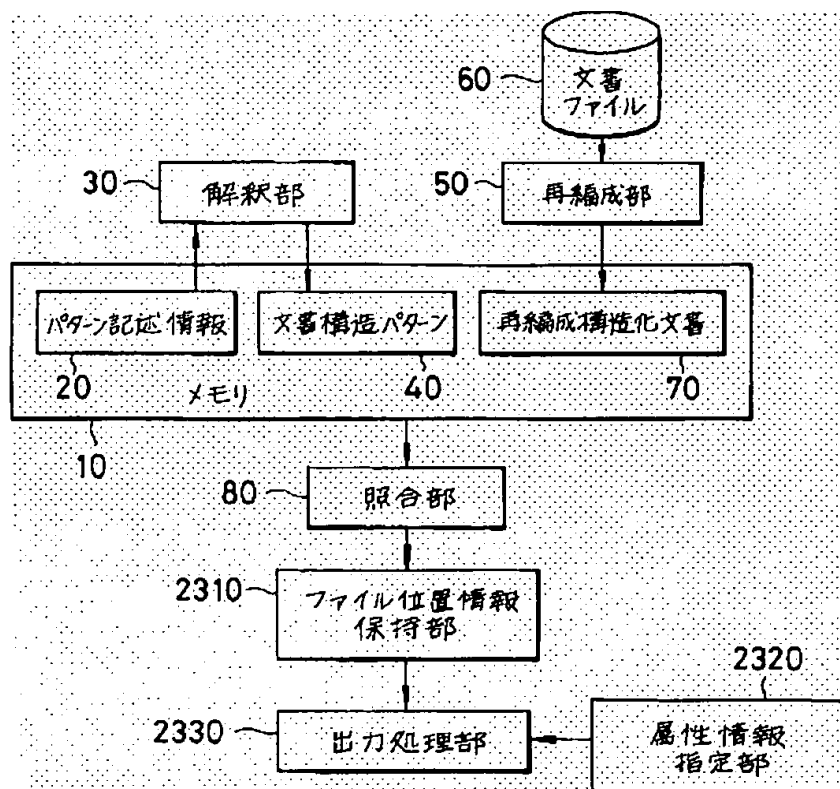
2007: The extracted partial structure was inserted in the inserting point shown by B [j].

2009: The result was outputted.

End



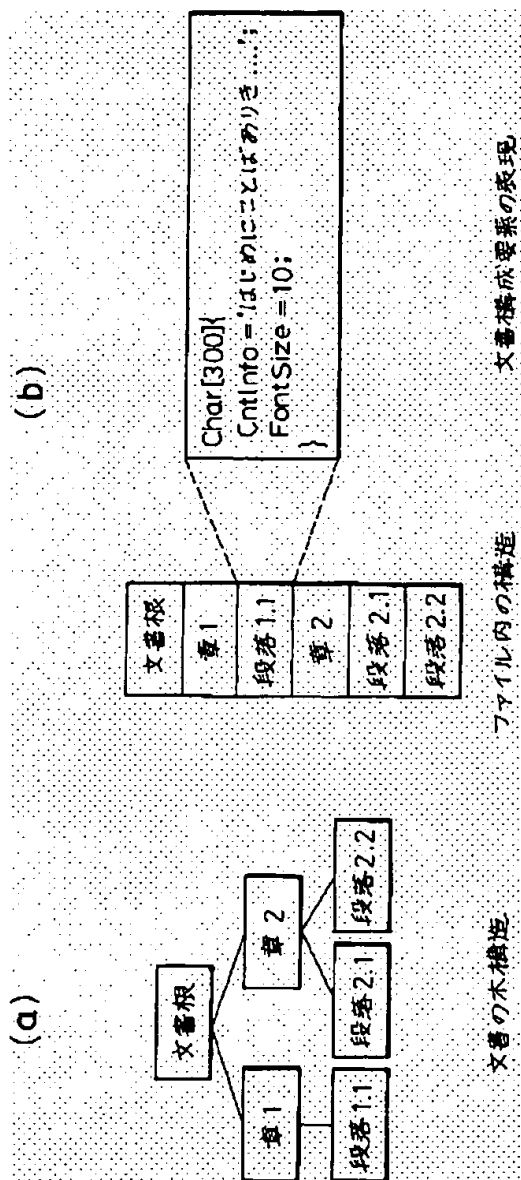
[FIG 28]



- 60 Text file
- 30 Interpretation section
- 50 Reorganization section
- 20 Pattern description information
- 40 Document structure pattern
- 70 Reorganization structuring document
- 10 Memory
- 80 Collation section
- 2310: File positional-information retainer
- 2330: Output-processing section
- 2320: Attribute information designation section.

【図 29】

[FIG 29]



(a)

Document solution

Chapter 1

Chapter 2

Paragraph 1.1

Paragraph 2.1

Paragraph 2.2

The tree structure of a document

(b)

Document solution

Chapter 1

Paragraph 1.1

Chapter 2


Paragraph 2.1

Paragraph 2.2

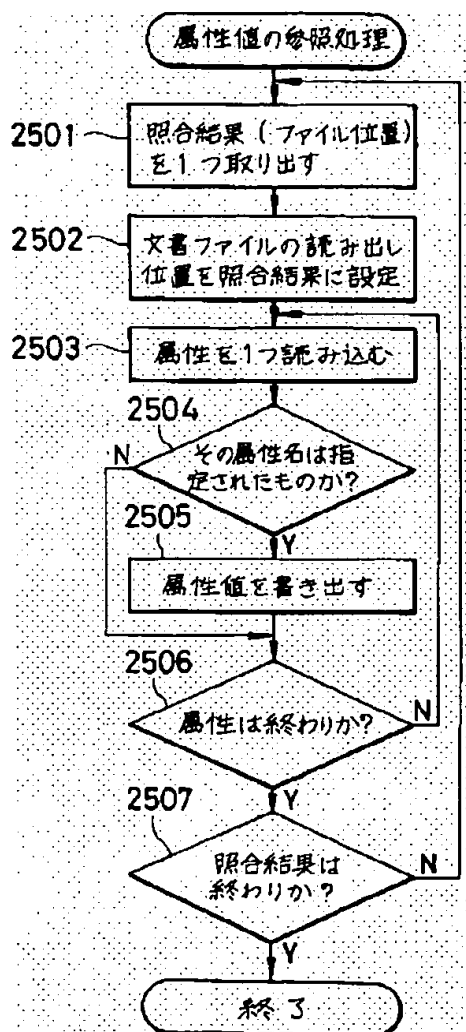
Structure in a file

Char[300]{Cnt Info='There was language first... '; Font Size=10;}

Expression of a document component

【 3 0】

[FIG. 30]



Refer processing of attribute value

2501: Taking out one collation result (file position).

2502: The read-out position of a text file was set as the collation result.

2503: Reading one attribute.

2504: Wasn't the attribute name designated?

2505: Writing out attribute value.

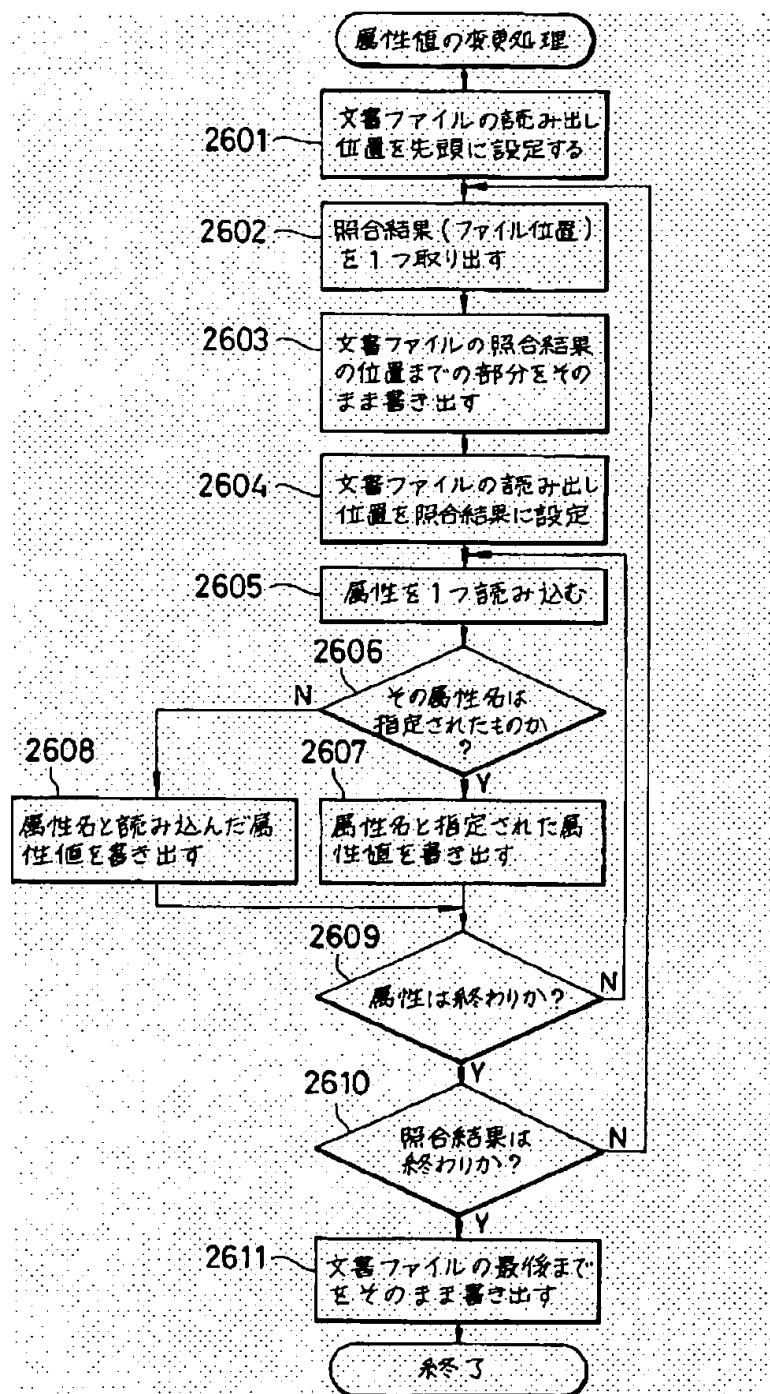
2506: Is an attribute end?

2507: Is a collation result end?

Completion

【図 31】

[FIG 31]



Alteration processing of attribute value

2601: Setting the read-out position of a text file as a head.

2602: Taking out one collation result (file position).
2603: Writing out the part to the position of the collation result of a text file as it is.
2604: The read-out position of a text file was set as the collation result.
2605: Reading one attribute.
2606: Wasn't the attribute name designated?
2607: Writing out the attribute value designated as the attribute name.
2608: Writing out an attribute name and the read attribute value.
2609: Is an attribute end?
2610: Is a collation result end?
2611: Writing out even the last of a text file as it is.
Completion



DERWENT TERMS AND CONDITIONS

Derwent shall not in any circumstances be liable or responsible for the completeness or accuracy of any Derwent translation and will not be liable for any direct, indirect, consequential or economic loss or loss of profit resulting directly or indirectly from the use of any translation by any customer.

Derwent Information Ltd. is part of The Thomson Corporation

Please visit our home page:

["WWW.DERWENT.CO.UK"](http://WWW.DERWENT.CO.UK) (English)

["WWW.DERWENT.CO.JP"](http://WWW.DERWENT.CO.JP) (Japanese)